VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

# Digitální stopy, identita a její ochrana

# Digital Footprint, Identity and its Protection

2015

Daniel Žažo

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

# Zadání diplomové práce

Student: **Bc. Daniel Žažo**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Digitální stopy, identita a její ochrana
Digital Footprint, Identity and its Protection

Zásady pro vypracování:

Práce se zabývá digitální identitou a její ochranou. Cílem je vytvořit přehledovou studii existujících ochranných metodách, provedení analýz a navržení vlastní ochranné metody.

1. Seznamte se s problematikou digitálních stop a identity.
2. Porovnejte existující metody ochrany a zcizení.
3. Určete vhodné charakteristiky použitelné pro detekci a ochranu zjištěných digitálních stop.
4. Navrhněte a implementujte aplikaci pro ochranu před monitorováním digitálních stop.
5. Diskutujte dosažené výsledky a navrhněte možná vylepšení.

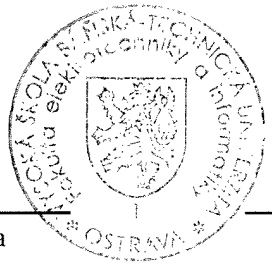
Seznam doporučené odborné literatury:

[1] Merhaut F., Zelinka I., Úvod do počítačové bezpečnosti, Fakulta aplikované informatiky, UTB ve Zlíně, Zlín, 2009
[2] Peter Szor, Počítačové viry - analýza útoku a obrana, Zoner Press, ISBN 80-86815-04-8, 2006
[3] Pokorný J., Hacking - umění exploitace, Zoner Press, ISBN: 978-80-7413-022-9, 2009
[4] Lance J., Phishing bez záhad, Grada 2007, ISBN: 80-247-1766-2
[5] Garfinkel, Simson; Cox, David. "Finding and Archiving the Internet Footprint". Presented at the first Digital Lives Research Conference. London, England. 2009, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.149.4193

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **prof. Ing. Ivan Zelinka, Ph.D.**

Datum zadání: 01.09.2013
Datum odevzdání: 07.05.2015




doc. Dr. Ing. Eduard Sojka
*vedoucí katedry*

prof. RNDr. Václav Snášel, CSc.
*děkan fakulty*

**Prohlášení studenta**

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 20. dubna 2015

Daniel Žažo

**Poděkování**

Na tomto místě bych rád poděkoval vedoucímu diplomové práce panu prof. Ing. Ivanu Zelinkovi, Ph.D. za podnětné vedení této práce. Jeho rady mi velice pomohly při zdokonalování práce. Dále bych chtěl poděkovat všem, kteří mě podporovali. Děkuji.

## Abstrakt

Popularita internetu s přibývajícím časem narůstá a stává se součástí stálé více domácností. Nejedná se ovšem jen o mladé generace, co si internet pořizují, ale lze zde zařadit osoby všech věkových kategorií. At' už je důvod pořízení jakýkoliv, tak mezi těmito osobami jsou takoví, kteří s internetem nemají moc zkušeností. V horším případě s ním nemají zkušenost žádnou a právě takových lidí je na internetu většina, čehož zneužívají podvodníci a zloději z nejrůznějších oblastí. Obzvlášť lidé, co o sobě zveřejňuji mnoho informací na sociálních sítích a ne jenom tam, jsou potenciálním cílem útočníků.
Cílem této práce je ukázat jaká nebezpečí se na internetu skrývají a jak zabránit případným nepříjemnostem, které vznikají především zveřejňováním jakýchkoliv informací, které se obecně nazývají *digitální stopa*. Součástí této je práce je taky implementace vlastní aplikace (internetový robot) na monitorování *digitálních stop* a aplikace, na ochranu před monitorováním *digitálních stop*.

**Klíčová slova:** anonymita na internetu, digitální stopa, ztráta soukromí, krádež identity, bezpečnost na internetu, internetový bot/robot, crawler, vyhledávání, monitoring, java

## Abstract

The Internet is growing in popularity as time goes by and in becoming a part of more and more households. It is not just about a young generation that in buying the Internet but people of all ages can be included in here. Whatever the reason for any acquisition is, there are those among these people who do not have much experience with the Internet. In the worn case they have no experience with the Internet. and there very people create the majority on the Internet and are abused by crooks and thieves from various fields. People who publish a lot of personal information on social networking sites and not only there to be a potential target for attackers.
The aim of this work is to show what dangers hide in the Internet and how to prevent any possible inconvenience ariving primarily from publishing any information that in generally known as a *digital footprint*. Part of this work is also the implementation of custom applications (web robot) for monitoring *digital footprint* and application for protection from the monitoring of *digital footprint*.

**Keywords:** anonymity on the Internet, digital footprint, loss of privacy, identity theft, Internet security, internet bot/robot, crawler, search, monitoring, java

## Seznam použitých zkratek a symbolů

| | | |
|---|---|---|
| ITU | – | International Telecommunication Union |
| OSN | – | Organizace spojených národů |
| TCP/IP | – | Transmission Control Protocol/Internet Protokol |
| IP | – | Internet Protokol |
| IT | – | Informační technologie |
| ICANN | – | Internet Corporation for Assigned Names and Numbers |
| RIR | – | Regional Internet Regostry |
| AfriNIC | – | African Network Information Center |
| APNIC | – | Asia Pacific Network Information Centre |
| ARIN | – | American Registry for Internet Numbers |
| LACNIC | – | Latin America and Caribbean Network Information Centre |
| RIPE NCC | – | Réseaux IP Européens Network Coordination Centre |
| LIR | – | Local Internet Registry |
| CESNET | – | Czech Education and Scientic NETwork |
| UPC | – | United Pan-European Communications |
| O2 | – | Telefónica O2 |
| ISP | – | Internet Service Provider |
| WHOIS | – | Who is? |
| FTP | – | File Transfer Protocol |
| SSID | – | Service Set Identification |
| MAC | – | Media Access Controll |
| BSSID | – | Basic Service Set Identification |
| BTS | – | Base Transceiver Station |
| GPS | – | Global Positioning System |
| API | – | Application Programming Interface |
| W3C | – | World Wide Web Consortium |
| MIME | – | Multi-Purpose Internet Mail Extensions |
| PHP | – | Hypertext Preprocessor |
| WWW | – | World Wide Web |
| HTTP | – | Hypertext Transfer Protocol |
| HTTPS | – | Hypertext Transfer Protocol Secure |
| TOR | – | The Onion Router |
| NSA | – | National Security Agency |
| JAP | – | Java Anon Proxy |
| VPN | – | Virtual Private Network |
| P2P | – | Peer-to-peer |
| I2P | – | Invisible Internet Project |
| SSL | – | Secure Sockets Layer |

| | | |
|---|---|---|
| SMTP | – | Simple Mail Transfer Protocol |
| OCR | – | Optical Character Recognition |
| IRC | – | Internet Relay Chat |
| ICQ | – | I Seek You |
| DoS | – | Denial of Service |
| DDoS | – | Distributed Denial of Service |
| HTML | – | HyperText Markup Language |
| API | – | Application programming interface |
| IDS | – | Intrusion detection system |
| WAR | – | Web application ARchive |

# Obsah

# Seznam tabulek

# Seznam obrázků

# 1 Úvod

V dnešní době je počítač nemyslitelnou součástí mnoha lidí na světě, a to především ve vyspělejších zemích. Položit zde někomu otázku, zda vlastní počítač a slyšet kladnou odpověď, už snad nikoho nepřekvapí. Překvapením by bylo slyšet odpověď opačnou, ale tu jen tak neuslyšíme. Používání počítače je dnes spjato v drtivé míře s internetem, který už je dostupný takřka všude, a to nejen pro počítače, ale i pro mobilní zařízení (notebooky, mobilní telefony, tablety, různé herní konzole, aj.). V mnoha případech může být internet hlavním důvodem, proč si počítač vůbec pořizovat, ať už je to z důvodů pracovních, za účelem vzdělávání nebo jen pro zábavu.

Používání internetu je spjato hlavně s klasickým stolním počítačem nebo notebookem, protože uživatel si může řadu problémů a otázek vyřešit snadným způsobem ovládání. Neví, v kolik mu jede vlak do práce? Neví, jaký má peněžní zůstatek na bankovním účtu? Neví, jaké mají jeho děti známky ve škole? Neví, jaké je hlavní město Filipín? Zkrátka pokud uživatel dokáže využít potenciál internetu, může dostat odpověď, na jakoukoliv otázku, která ho napadne. Pokud se uživatel chce zabavit, tak to taky není žádný problém, protože na internetu může hrát hry, chatovat a dělat cokoliv, co je pro něj zábavné.

Pozadu nezůstávají ani výrobci různé elektroniky, u které má využití internetu smysl. Například novějším televizím s funkcí *Smart*, které mají zabudované rozhraní pro připojení internetu, stačí už jen nějaký zabudovaný internetový prohlížeč a majitel si může vesele brouzdat po internetu, tak jako by byl na počítači. Ovládání sice nebude úplně ideální, ale funkčnost je úplně stejná. To znamená, že uživatel nemusí jen navštěvovat internetové stránky, ale může si pouštět videa, přispívat do diskuzí, přihlašovat se do emailových schránek a vše, co mu onen zabudovaný prohlížeč umožní. Součástí těchto televizí nebývá jen internetový prohlížeč, ale sada základních zabudovaných aplikací používající internet, které poskytují využití pro nejrozšířenější internetové služby, jakými jsou například *Youtube* nebo *Facebook*. Pomocí těchto zabudovaných aplikací lze potom snadněji danou službu využívat (např. vkládání komentářů, vyhledávání, aj.).

Jak je vidno, tak internet přináší mnoho kladného využití, jenž uživatel opravdu ocení z mnoha různých hledisek. Bohužel, jak už to ve světě bývá, všechno, co má klady, má i zápory a internet v tomto není výjimkou. Uživateli přináší mnoho hrozeb a jednou z nich je *digitální stopa*, kterou po sobě zanechává téměř každý, kdo internet použije. Výjimku může tvořit ten uživatel, který si nepřeje být identifikován nebo lokalizován při použití jakékoliv internetové služby. Nezanechat za sebou *digitální stopu* je ovšem úkol pro pokročilejší uživatele, protože někdy není na internetu viditelně zanechána, ale ukládá se do různých (např. logovacích) souborů, ať už o tom uživatel ví nebo ne. Konkrétní data pak představuje vlastní obsah souboru nebo jeho metadata. *Digitální stopu* představují také soubory, které jsou na jakémkoliv počítači a nemusí se jednat o soubory vytvořené na internetu, jak by se mohlo na první pohled zdát. Zkušený uživatel si tuto potencionální hrozbu uvědomuje a dává si pozor, ale ten lehkomyslnější to nemusí brát vážně. V horším případě o *digitální stopě* vůbec neslyšel. Ve skutečnosti ale může být i ten sebe-opatrnější uživatel poškozen.

Nyní uvedu jeden z mnoha možných případů, kdy se dá *digitální stopa* proti někomu

použít. Pro tento případ budu potřebovat smyšlenou osobu, kterou pojmenuji např. Jan Novák. Ten si hledá novou práci a našel inzerát, který je pro něho jako stvořený. Ihned tedy kontaktuje společnost, jež tento inzerát vytvořila. V dané společnosti personalisté jako první krok požadují po panu Novákovi zaslání jeho životopisu s trochou základních informací. Pán Novák tedy neváhá a obratem zasílá svůj životopis. Ten je následně důkladně prostudován a výsledkem je, že pan Novák je dle životopisu ideálním kandidátem na hledanou pozici. U následného pohovoru může tedy mít velkou šanci, ale ještě před jeho pozváním si ho proklepnou. To znamená, že personalista zabývající se náborem nových zaměstnanců zkusí dohledat na internetu vše, co je spjato s panem Novákem. Tento proces přece nic nestojí a ve firmách se stává víc a víc populárnější. Co kdyby byl uchazeč o zaměstnání velký milovník alkoholu? Bylo by z pohledu společnosti rozumné takového člověka přijmout? Pravděpodobně ne a přednost by dostal ten, o kterém by tyto informace nebyly zjištěny. Pro vyhledávání jsou použity informace uvedené v životopise, především jméno a bydlištěm. Ze zbývajících informací lze výsledky upřesnit a rozšířit. Pro dohledání informací může být využito dvou možností.

První možností je použít internetové vyhledávače. Vyhledávání netrvá dlouho a na sociálních sítích byl nalezen profil pana Nováka, kde je zveřejněno poměrně mnoho informací. Má tam uvedeno datum narození se svým bydlištěm včetně fotografie, kterou má taky v životopise, takže o spojení tohoto profilu s uchazečem o zaměstnání už není velkých pochyb. Personalista provádějící vyhledávání se bude chtít dále podívat na fotky pana Nováka, ale bohužel má fotky přístupné jen přátelům a tou personalista není. Nevadí, tak se zkusí podívat na pár jeho přátel a jeden z nich má fotky přístupné. Na některých z nich je i pan Novák a to v situacích (např. popíjení alkoholu), které by nejraději vymazal ze svého života. Na základě těchto fotek se personalista rozhodne pana Nováka vyškrtnout ze seznamu zájemců. V jednodušším scénáři má pan Novák fotky přístupné a na nich je například vidět, že propaguje určitou politickou stranu, která je pravým opakem té, ve které působí právě onen personalista, který se panem Novákem zabývá. A to může být důvodem, proč tato osoba pana Nováka taktéž vyškrtne ze seznamu zájemců o nabízenou pozici. Pan Novák pak jen obdrží zprávu, že o něj zájem nemají, a to na základě již smyšlených důvodů, ale to se pan Novák nikdy nedozví.

V druhém případě lze pro nalezení informací použít *vyhledávací robot (crawler, spider)*, což je speciální program, zpracovávající internetové stránky. Tento robot prohledává jednotlivé stránky a najde-li nějaké odkazy, tak stránky pod těmito taktéž prohledá. Ve výsledku lze sestavit mapy různých webů nebo jiných zajímavých informací, zobrazující účel, ke kterému byl robot stvořen. V tomto případě bude při procházení stránek vyhledáván navíc pan Novák a bude-li mít osoba provádějící nábor štěstí, tak se dozví, jaké internetové stránky navštívil a co na nich dělal (např. komentáře). I v tomto případě může být o panu Novákovi nalezeno něco, co rozhodne o jeho vyškrtnutí ze seznamu zájemců o nabízenou práci a pan Novák nebude přijat.

Co víc mohl pan Novák udělat, aby k tomuto nedošlo? Jednou možností by bylo nezakládat si profily na sociálních sítích nebo si je lépe zabezpečit a nevyplňovat každou kolonku, která je po něm požadována. Každopádně nejbezpečnější by bylo, kdyby internet vůbec nepoužíval nebo se choval alespoň obezřetně, protože tak by ho nikdo nikde

nemohl vyhledat, pokud by jeho profil někdo fiktivně nezaložil. Na uvedeném příkladu je vidět, jak málo stačí, aby *digitální stopa* mohla být proti někomu zneužita. Pokud se ale podíváme na uvedený příklad z pohledu společnosti hledající nového zaměstnance, pan Novák není díky *digitální stopě* do společnosti přijat jakožto potencionální problematická osoba. Ať už to ve skutečnosti pravda je či není. Tyto procesy o zjišťování informací na internetu o osobách lze zařadit do kategorie *social engineering*, což je způsob získávání informací na internetu za určitým cílem.

S *digitální stopou* na internetu je úzce spjata taktéž *identita* uživatele. Na tu by měl každý uživatel internetu dávat pozor, protože s ní přichází další rizika, jakými jsou např. ztráta soukromí či zneužití uživatelovy *identity*. O této problematice taky nemá mnoho uživatelů internetu povědomí a myslí si, že internet je určitým způsobem anonymní, což ve skutečnosti není pravda. Existuje taky možnost, že někdo by chtěl poškodit určitou osobu. V tomto případě může být útočníkem založen na nějaké sociální síti profil se jménem dané osoby a pokud má i nějakou fotografii, tak ji k profilu přidat. Takto založený profil potom bude působit velmi věrohodně. Útočník pak může zneužívat *identitu* osoby k poškozování jeho jména nebo za jiným kompromitujícím účelem. Až daná osoba zjistí, že někdo za jejími zády páchá škodu pod jejím jménem, tak bude nemile překvapena, jak očištění jména není jednoduchá záležitost. Založením fiktivního profilu samozřejmě útočník neukradne cizí osobě peníze z bankovního účtu, protože to už podléhá vyššímu zabezpečení a ověřování, ale mohl by ho velmi poškodit.

Jak takovému útoku může někdo zabránit? Kdo zabrání někomu vytvořit profil za použití cizího nebo smyšleného jména na různých sociálních sítích? Odpověď je zcela jasná! Tomu nezabrání nikdo, takže kdykoliv a kdokoliv může být na internetu poškozen. Pokud by už pachatel chtěl někoho poškodit, tak musí počítat s tím, že za to může být dopaden a třeba i trestně stíhán, protože internet není anonymní, jak si mnozí myslí a každý za sebou zanechává určitou *digitální stopu*. Podle statistik od *Internet Live Stats* [1], které měří počet připojených uživatelů k internetu na základě zpracování údajů *Mezinárodní telekomunikační unie (ITU)* a *Populační divizí OSN*, je na světě k 1. červenci připojených 2 925 249 355 uživatelů z celkových 7 243 784 121 obyvatel světa a toto číslo narůstá každou sekundu. Z uvedených čísel vyplývá, že k internetu je připojeno více než 40% lidské populace a pro všechny tyto uživatele je *digitální stopa* velkou hrozbou.

# 2 Anonymita v prostředí internetu

Hrozba *digitální stopy* je spjata ve velké míře s internetem, o kterém si většina uživatelů myslí, že je to svět anonymní nebo je anonymní alespoň do jisté míry. Hned na začátku této práce bych chtěl tuto špatnou domněnku vyvrátit a uvést na pravou míru, protože pohyb na internetu anonymní opravdu není. Každé zařízení připojené k internetu musí dodržovat řadu pravidel bez ohledu na to, o jaké zařízení se jedná a z úvodu 1 už víme, že v dnešní době jich není málo.

## 2.1 IP adresy

Pokud v počítačové síti mezi sebou chtějí komunikovat dvě zařízení, tak musí mít jednoznačné identifikátory pro jejich přesné vyhledání a následnou komunikaci. Internet je založený na rodině protokolů *TCP/IP* a v důsledku využívání tohoto protokolu internet používá jako jednoznačný identifikátor zařízení v síti *IP adresu*. Ta jednoznačně identifikuje síťové rozhraní v počítačové síti, která používá *IP (internet protocol)*. Existují dvě verze *IP adresy*.

- **IPv4**
  Aktuálně nejrozšířenější verze *IP*, která používá 32-bitové adresy. Umožňuje tedy vytvořit $2^{32}$ = 4 294 967 296 adres. Část adres je určena pro interní potřeby protokolu. Příklad *IPv4* je *47.16.255.1*.

- **IPv6**
  Postupně nahrazuje *IPv4* z důvodu nedostatku adresovatelného počtu adres. Tato verze už umožňuje adresovat $2^{128} = 3,4 * 10^{38}$ adres, což je obrovské číslo, které už vystačí na mnohem delší dobu.

Aby *IP adresy* byly lépe zapamatovatelné a uživatel nemusel zadávat do prohlížeče složitou číselnou kombinaci, existuje systém doménových jmen *(DNS)*, který námi zadanou adresu překládá na *IP adresu* (např. *www.vsb.cz [6] na http://158.196.149.111/)*, což je pro každého uživatele mnohem přívětivější způsob na zapamatování si internetových stránek.

Každé zařízení v prostředí internetu komunikuje na základě *IP adres* a aby mezi sebou mohly komunikovat dvě zařízení, tak tyto *IP adresy* musí být veřejné. Vypadá to tedy tak, že každý uživatel má veřejnou *IP adresu*, protože se dostane na každou internetovou stránku, na kterou chce. To by znamenalo, že *IP adresy* by měly být v dnešní době skoro všechny zarezervovány, ale nejsou. Ve skutečnosti je to řešeno tak, že poskytoval internetu tzv. *provider*, má veřejnou *IP adresu* a všechny své zákazníky má ve své privátní síti, takže pokud zákazník vstupuje na internet, připojuje se tak přes bránu od *providera* a zákazník se na internetu tváří jako uživatel s veřejnou *IP adresou*. To by ale znamenalo, že s uživatelem v privátní síti *providera* nemůže nikdo komunikovat, protože ho nevidí. Ve skutečnosti ale komunikuje a tuto komunikaci už zajišťují síťové prvky, které si pamatují, který uživatel kam přistupuje. Odpověď dotazovaného zařízení pak směrují přímo ke konkrétnímu uživateli. Bude-li chtít někdo s veřejnou *IP adresou* komunikovat se zařízením s privátní

*IP adresou*, tak to možné není. Někdy jsou ale případy, kdy někdo potřebuje veřejnou *IP adresu*, protože chce provozovat nějakou službu (např. *FTP server*) a musí si veřejnost *IP adresy* u *providera* zajistit pravděpodobně za nějaký poplatek. Základní vlastnosti typu těchto adres jsou:

- **Privátní (neveřejné)**
  - **Výhody**
    - Ochrana před *viry* a *crackery* (odborníci s vysokou znalostí fungování systémů, jež narušují počítačové sítě), protože ochranu zajišťuje *provider.*
    - Anonymita na internetu. Platí jen do jisté míry, protože uživatel na internetu není nikdy anonymní. Minimálně *provider* ví o uživateli.
  - **Nevýhody**
    - Zařízení není viditelné z venku, a proto nelze provozovat některé síťové služby (např. *FTP*).
- **Veřejné**
  - **Výhody**
    - Přímá dostupnost zařízení (počítače) z venku.
    - Možnost provozovat veřejné síťové služby (např. *FTP*).
    - Možnost hraní síťových her, které vyžadují veřejnou *IP adresu*.
  - **Nevýhody**
    - Ztráta ochrany proti *virům* a útokům *crackerů*, a proto je potřeba mít dobře zabezpečené síťové zařízení různými *firewally*.
    - Ztráta anonymity, jelikož je zcela jasné, kdo kam přistupuje.
    - Pro nezkušené uživatele je to riziko.

Výhoda privátní adresy je anonymita na internetu, ale není to pravda v případech, kdy by mohl být v dané zemi porušen zákon. Bezpečnostní složka státu (policie) si může u *providerů* vyžádat historii o tom, kdo, kdy a kde byl připojen, protože ti ji musí určitý čas uchovávat.

*IP adresu* je možno klasifikovat taky jako *statickou* a *dynamickou*. U *statické IP adresy* má zákazník zaručeno, že po každém připojení k internetu bude mít *IP adresu* stejnou. To je potřeba zajistit, pokud chce zákazník provozovat veřejně dostupné služby. Používá-li zákazník od *providera dynamickou IP adresu*, tak má většinou přidělenou jednu *IP adresu* napevno a může to vypadat, že ji má statickou, ale v případě té dynamické to není zaručeno.

## 2.2 Přidělování IP adres

O přidělování *IP adres* se stará organizace *ICANN*, která má pod kontrolou celý adresový rozsah a zajišťuje, aby byla dodržena jejich unikátnost. Tato korporace rozděluje adresové bloky regionálním internetovým registrům *(RIR)*, které si svět rozdělily na pět přibližně stejně velkých částí, odpovídající jednotlivým kontinentům.

1. **AfriNIC** - Afrika
2. **APNIC** - Jihovýchodní Asie a Austrálie
3. **ARIN** - Severní Amerika, část Karibiku a severoatlantické ostrovy
4. **LACNIC** - Jižní a střední Amerika a část Karibiku
5. **RIPE NCC** - Evropa, Blízký východ a centrální Asie


Obrázek 1: Rozdělení světa na jednotlivé regionální internetové registry [5]

Všichni tito registrátoři pokrývají obrovskou oblast, proto přidělené bloky dále poskytují tzv. lokálním internetovým registrům *(LIR)*. Tito lokální registrátoři jsou už většinou jednotliví provideři *(CESNET, UPC, O2, atd.)* každého uživatele tzv. *internet service provideři (ISP)*.

## 2.3 Dostupné informace

### 2.3.1 IP adresa/doménové jméno - základní informace

Mnoho informací o každé existující *IP adrese* je veřejně dostupné v tzv. *WHOIS databázích*, kde se může kdokoliv a kdykoliv na internetu podívat, komu určitá *IP adresa* patří.

Z *IP adresy* se sice nedá zjistit uživatelovo jméno nebo přesná adresa jeho bydliště, ale mezi zjištěnými informacemi budou údaje především od *providera*. Pokud je ale žádáno o více *IP adres*, o které většinou žádají různé firmy, tak pak přichází podmínka přímo od koordinačního centra *RIPE NCC*, které stanovuje pravidlo, že ve *WHOIS databázích* musí být uvedena adresa zájemce, kterou je pak možno dohledat z přidělených *IP adres*. Jednotlivcům ale nehrozí, že z jeho *IP adresy* po vyhledávání ve *WHOIS databázích* bude nalezena jeho adresa. Tu si uchovává jen konkrétní uživatelův *provider*.

Na internetu existuje více internetových stránek, kde je možné dohledat informace o *IP adrese*. Tyto stránky získávají informace z centrální *databáze WHOIS* a v jednotlivých případech se jedná pouze o jinou prezentaci získaných výsledků. Velice dobře funguje web *https://who.is/* [7], který si poradí s *IP adresu* i doménovým jménem a já jsem zde s dohledáním potřebných informací taktéž neměl problém. Pro uvedení příkladu jsem si ale vybral českou internetovou stránku *http://www.nic.cz/whois/* [8], jež některé zkoušené *IP adresy* nevyhledala. Výsledek zde je ale velice přehledný a chtěl jsem demonstrovat, že ve *WHOIS databázích* je možno vyhledávat i na českých stránkách. Po zadání *URL adresy vsb.cz* mi vyjedou následující informace a ty, které jsou ve výsledku podtrženy, jdou dále rozvinout a zobrazí se doplňující informace o konkrétní položce. Výsledek je následující.

**PROHLÍŽENÍ DOMÉNOVÉHO JMÉNA**
**Doménové jméno** - vsb.cz
**Registrace od** - 05.11.1995
**Poslední aktualizace** - 04.04.2012 03:18:17
**Datum expirace** - 20.10.2015
**Držitel** - SB:PT7_XX Vysoka skola banska - Technicka univerzita Ostrava
**Administrativní kontakt** - PT7 Martin Pustka
**Dočasný kontakt** - (neuvedeno)
**Určený registrátor** - REG-MIRAMO MIRAMO spol.s r.o. od 20.10.2004 09:35:00
**Zabezpečeno pomocí DNSSEC** - no
**Stav** - (neuvedeno)
**Sada jmenných serverů** - NSS:PT7_XX:2
**Jmenný server** - decsys.vsb.cz 158.196.149.9
**Jmenný server** - ns.ces.net
**Jmenný server** - sun.uakom.sk
**Technický kontakt**
- SB:PT7_XX Vysoka skola banska - Technicka univerzita Ostrava
- JGRYGAREK Jiri Grygarek
- PUMA Martin Pustka
- PT7 Martin Pustka

**Určený registrátor** - REG-MIRAMO MIRAMO spol.s r.o. od 01.10.2007 02:00:00
**Stav** - Je navázán na další záznam v registru

---

**PROHLÍŽENÍ DOMÉNOVÉHO JMÉNA** - SB:PT7_XX
**Identifikátor** - SB:PT7_XX

**Organizace** - Vysoka skola banska - Technicka univerzita Ostrava
**Jméno** - Vysoka skola banska - Technicka univerzita Ostrava
**DIČ** - CZ61989100
**Typ identifikace** - IČO
**Identifikační údaj** - 61989100
**E-mail** - martin.pustka@vsb.cz
**E-mail pro oznámení** - Martin.Pustka@vsb.cz
**Telefon** - +420.596993174
**Fax** - (neuvedeno)
**Registrace od** - 30.08.2002
**Vytvořeno registrátorem** - REG-CZNIC CZ.NIC, z.s.p.o.
**Poslední aktualizace** - 22.05.2012 16:24:31
**Poslední transfer** - (neuvedeno)
**Adresa** - 17.listopadu 15/2172, 708 33, Ostrava-Poruba, CZ
**Určený registrátor** - REG-MIRAMO MIRAMO spol.s r.o.
**Stav** - Je navázán na další záznam v registru

---

**PROHLÍŽENÍ KONTAKTU** - PT7
**Identifikátor** - PT7 + ikona *Založit MojeID*
**Organizace** - (neuvedeno)
**Jméno** - Martin Pustka
**DIČ** - (neuvedeno)
**Typ identifikace** - (neuvedeno)
**Identifikační údaj** - (neuvedeno)
**E-mail** - martin.pustka@vsb.cz
**E-mail pro oznámení** - Josef.Verich@vsb.cz
**Telefon** - +420.597323174
**Fax** - +420.596919352
**Registrace od** - 10.08.2001
**Vytvořeno registrátorem** - REG-CZNIC CZ.NIC, z.s.p.o.
**Poslední aktualizace** - 27.10.2013 08:38:17
**Poslední transfer** - (neuvedeno)
**Adresa** - 17.listopadu 15/2172, 708 33, Ostrava-Poruba, CZ
**Určený registrátor** - REG-MIRAMO MIRAMO spol.s r.o.
**Stav** - Je navázán na další záznam v registru

---

**PROHLÍŽENÍ KONTAKTU** - JGRYGAREK
**Identifikátor** - JGRYGAREK + ikona *Založit MojeID*
**Organizace** - (neuvedeno)
**Jméno** - Jiri Grygarek
**DIČ** - (neuvedeno)
**Typ identifikace** - (neuvedeno)
**Identifikační údaj** - (neuvedeno)

**E-mail** - Jiri.Grygarek@vsb.cz
**E-mail pro oznámení** - Josef.Verich@vsb.cz
**Telefon** - +420 596993240 ;+420 596919352
**Fax** - +420 596919352
**Registrace od** - 30.08.2002
**Vytvořeno registrátorem** - REG-CZNIC CZ.NIC, z.s.p.o.
**Poslední aktualizace** - (neuvedeno)
**Poslední transfer** - (neuvedeno)
**Adresa** - 17.listopadu 15/2172, 708 33, Ostrava-Poruba, CZ
**Určený registrátor** - REG-MIRAMO MIRAMO spol.s r.o.
**Stav** - Je navázán na další záznam v registru

---

**PROHLÍŽENÍ KONTAKTU** - PUMA
**Identifikátor** - PUMA + ikona *Založit MojeID*
**Organizace** - (neuvedeno)
**Jméno** - Martin Pustka
**DIČ** - (neuvedeno)
**Typ identifikace** - (neuvedeno)
**Identifikační údaj** - (neuvedeno)
**E-mail** - Martin.Pustka@vsb.cz
**E-mail pro oznámení** - Martin.Pustka@vsb.cz
**Telefon** - +420.596993174
**Fax** - (neuvedeno)
**Registrace od** - 30.08.2002
**Vytvořeno registrátorem** - REG-CZNIC CZ.NIC, z.s.p.o.
**Poslední aktualizace** - 22.05.2012 16:24:31
**Poslední transfer** - (neuvedeno)
**Adresa** - 17.listopadu 15/2172, 708 33, Ostrava-Poruba, CZ
**Určený registrátor** - REG-MIRAMO MIRAMO spol.s r.o.
**Stav** - Je navázán na další záznam v registru

---

**PROHLÍŽENÍ REGISTRÁTORA** - REG-MIRAMO
**Identifikátor** - REG-MIRAMO
**Jméno** - MIRAMO spol.s r.o.
**Telefon** - (neuvedeno)
**Fax** - (neuvedeno)
**URL** - http://www.9net.cz
**Adresa** - Albrechtičky 69, 74255, Albrechtičky

---

**PROHLÍŽENÍ REGISTRÁTORA** - REG-CZNIC
**Identifikátor** - REG-CZNIC
**Jméno** - CZ.NIC, z.s.p.o.
**Telefon** - +420 222 745 111

**Fax** - +420 222 745 112
**URL** - http://www.nic.cz
**Adresa** - Americká 23, 120 00, Praha 2

Z pouhého doménového jména (ve skutečnosti *IP adresa*) jsem zjistil poměrně mnoho informací týkající se především kontaktů na administrátory domény *vsb.cz* a také kdo jim danou doménu poskytl. Přesto už máme k dispozici adresu školy, emaily na administrátory a různé telefonní čísla, pomocí kterých by bylo možné dohledat na internetu další informace. Normálnímu uživateli jsou tyto informace k ničemu a moc mu neřeknou, ale pak existují zase takoví uživatelé, kteří s těmito informacemi dokážou pracovat a v horším případě je i zneužít.

### 2.3.2 IP adresa/doménové jméno - geolokalizace

Podle *IP adresy* není možné uživatele se 100% jistotou geolokalizovat, tedy není možné zjistit, kde se nachází zařízení, které je využíváno pro připojení k internetu. Přesto lze na internetu najít mnoho aplikací, podle kterých to údajně možné je.

Při zkoušení několika těchto internetových aplikací jsem byl vždy v Ostravě a ve většině případů jsem byl lokalizován do Moravskoslezského kraje, což je z pohledu Ostravy geograficky správně. V ostatních případech jsem byl geolokalizován na nejrůznějších místech celé České republiky a někdy jsem se měl nacházet i v jiných zemích. S jistotou se tedy na tyto služby spolehnout nedá, ale pokud je to pro někoho důležité, tak si musí tyto aplikace otestovat a na základě úspěšnosti si pak vybrat tu nejlepší.

Z kapitoly o dostupných informacích získaných z *IP adresy* 2.3.1 jsou vidět veškeré informace, které jsou dostupné pro kohokoliv, kdo je bude chtít znát. V případě internetových stránek *www.vsb.cz*, což jsou stránky mé školy, je vidět adresu administrátorů, která je totožná s adresou školy. Jelikož se jedná o velkou instituci, tak požadovala od *providera* větší počet *IP adres* a s tím přichází i podmínka od *RIP NCC*, která požaduje uvádění žadatelovy adresy. V tomto případě lze konstatovat, že z *IP adresy* byla zjištěna geografická poloha. Škola je ale přece jen velká svým rozsahem a přesnost umístění může být stovky metrů ne-li kilometrů. Pokud se tedy jedná o žadatele, jež požadují větší počet *IP adres*, tak určitá šance pro geolokalizaci existuje, ale pokud se např. jedná o společnost, která má více poboček po celé republice nebo dokonce po celém světě, tak určení geografické polohy z tohoto hlediska nepřipadá v úvahu. Žádá-li o veřejnou jednu *IP adresu* jednotlivec, tak i v případě veřejné *IP adresy* nebude možné adresu žadatele zjistit. Po vyhledání v *databázi WHOIS* bude zobrazen opět kontakt na *providera* i s jeho adresou.

Za zmínku určitě stojí uvést geolokalizaci, která nepracuje pouze s veřejnou *IP adresou*, jak jsem již uvedl, ale existuje i následující možnost. Má-li uživatelovo zařízení *Wi-Fi* rozhraní a má ho zapnuté, tak podle ostatních přípojných *Wi-Fi* bodů v okolí lze uživatele geolokalizovat [9] mnohem přesněji. Pro tuto geolokalizaci se využívají názvy sítí *(SSID)*, *MAC adresy* přístupových bodů pro bezdrátové připojení (*BSSID*, což je skrytý název sítě a to zjistit tuto *MAC adresu* nezamezí) a síly signálů. Po nashromáždění těchto údajů se používá celosvětová databáze *Wi-Fi* přístupových bodů, kde je možno zjistit požadovaný

výsledek. U mobilních zařízení pak lze navíc využívat systém základnových stanic (*BTS*), které přenášejí rádiové signály. Princip těchto stanic je stejný jako u *Wi-Fi*, ale systém základnových stanic pracuje na mnohem větší vzdálenosti, a proto přesnost geolokalizace už není tak dobrá. Řádově se může jednat o stovky metrů až kilometrů. Navíc novější mobilní zařízení mají zabudované *GPS*, které fungují už s přesností v jednotkách metrů a fungují zcela nezávisle na uvedených příkladech.

Pro všechny uvedené techniky existuje už mnoho naprogramovaných knihoven (např. *The Google Geocoding API [10], W3C Geolocation API [11]*), které využívají různých kombinací těchto nabízených možností. Tyto knihovny pak programátoři můžou snadno využít ve svých aplikacích a zjišťovat, kdo se kde nachází. Takže např. při hraní her může být uživatel geolokalizován, aniž by o tom měl nejmenší tušení.

Uvedené techniky nezaručí pravdivý výsledek, protože funkčnost internetu má své specifické vlastnosti, které nejdou obejít, ale v případě, že se někdo na někoho zaměří, tak existuje možnost, že si zjistí, kde se v danou chvíli nachází i v řádech desítek metrů. Určitě nepravdivý výsledek nastane v případě použití připojení přes vzdálenou plochu, kdy se uživatel připojí k jinému počítači, který je připojen k internetu. Tento počítač pak bude využíván k brouzdání po internetu a veškeré údaje se budou stahovat k počítači, ke kterému je uživatel právě připojen, nikoli k počítači u kterého se fyzicky nachází. Samotný uživatel může být třeba na druhém konci světa.

### 2.3.3 Email

Většina uživatelů, která používá internet, přišla do styku s elektronickou poštou tzv. *emailem*. Aby ho uživatel mohl využít, musí mít svou *emailovou adresu*, kterou si může zcela bezplatně vytvořit. *Email* je jedna z nejdůležitějších elektronických služeb, protože je potřeba pro různé registrace, posílání soukromých zpráv a k mnoha dalším věcem, jež mají na internetu uživatelům usnadnit život. Každý, kdo si ho založil, tak s největší pravděpodobností nějaký ten email už taky odeslal nebo přijal. Už ale ne každý ví, že za každým takovým *emailem* se skrývá mnoho informací, ke kterým má přístup konkrétní odesílatel a příjemce. Běžný uživatel o těchto informacích ani neví, protože jsou skryté a navíc běžnému uživateli nic neříkají. Jsou uloženy v tzv. *hlavičce emailu (Email Headers)* [12], která doplní *email* před jeho odesláním a je jeho součástí. Po odeslání *email* koluje přes poštovní servery, které do hlavičky přidávají dodatečné informace. Celkem může být těchto informací poměrně mnoho a ve výsledku se jejich počet různí. Pro ukázku uvedu nejčastější z nich s jejich popiskem.

**EMAIL HEADERS**

**Received** - Cesta *emailu*, kde jsou zobrazeny doménové jména *(IP adresy)* v jaký čas a kudy email putoval.
**From** - *Emailová adresa* odesílatele.
**To** - *Emailové adresy* příjemců.
**Cc** - *Emailové adresy* příjemců v kopii.
**Bcc** - *Emailové adresy* příjemců ve skryté kopii.
**DKIM-Signature** - Elektronický podpis, kdy je ověřeno, že *emai*l je odeslán z

z pověřeného serveru, který je nastaven a ne z cizího, kdy by mohlo dojít k falšování.

**Reply-To** - *Emailová adresa* pro případnou odpověď.

**In-Reply-To** - Identifikace předcházející korespondence, která je odpovědí na mou zprávu.

**Subject** - Předmět *emailu*.

**Sender** - *Emailová adresa* odesílatele, pokud je jiná, než v položce *From*.

**Date** - Datum odeslání *emailu*.

**References** - Identifikace jiné korespondence, na kterou tato zpráva odkazuje.

**Message-ID** - *ID emailu*, které je automaticky generováno serverem.

**Keywords** - Klíčová slova.

**Comments** - Komentář.

**Content-type** - Určuje kódování obsahu a znakovou sadu.

**Mime-version** - Nastavení verze specifikace *MIME*.

**X-?** - Další parametry začínající písmenem X, které nejsou specifikovány v *MIME*, ale jsou používány jinými programy.

**X-Priority** - Např. 1, což představuje vysokou prioritu.

**X-Mailer** - *PHP* identifikace programu, který byl použit k odeslání *emailu*.

---

Jak je zřejmé, údajů v hlavičce je poměrně mnoho. Nyní uvedu příklad *hlavičky emailu* s přesnými hodnotami, kdy byl *email (aaa@centrum.cz)* odeslán od jednoho z největších poskytovatelů emailových schránek, a to od *www.centrum.cz* [13]. Příjemce tohoto *emailu* má *email (bbb@seznam.cz)* na konkurenčním serveru, který poskytuje emailové schránky a tím je *www.seznam.cz* [14]. Na tomto serveru jsem si následně nechal zobrazit hlavičku *emailu*, která obsahovala následující hodnoty.

**EMAIL HEADERS**

**Received** - from gmmr1.centrum.cz (gmmr1.centrum.cz [46.255.225.252])
by email-smtpd-v5.go.seznam.cz (Seznam SMTPD 1.2.89.1) with ESMTP;
Thu, 30 Oct 2014 09:25:15 +0100 (CET)

**Received** - from mail1006.cent (mail-g1.snat.cent [10.32.3.101])
by gmmr1.centrum.cz (Postfix) with ESMTP id 039CD80071CC
for <bbb@seznam.cz>; Thu, 30 Oct 2014 09:25:14 +0100 (CET)

**DKIM-Signature** - v=1; a=rsa-sha256; c=relaxed/relaxed; d=centrum.cz; s=mail;
by gmmr1.centrum.cz (Postfix) with ESMTP id 039CD80071CC
t=1414657514;
bh=1t7MR48cQlgYSQG46DwgRH/bPXBNL8Go5EE6qj+02O4=;
h=To:Subject:Date:From:From;
EuHgasdizDhf+g2LNguYkpKGBoFYLbFiEk45ouRM1mu3k3PL-
POpp6swlyMAcJlsnEa1/XDUoMvwv3+RvW05D9jlLFdmnwsD-
OskBdsnEC8P5JHtlk2HxxEZN1MPo1NjCoZLjvtu+PqFpB1ZtnV-
q7w1GtKeXltQHgZEBBsYy6Kpl5w=

**Received** - by mail1006.cent (Postfix, from userid 33)
id EBB3C6004A5F6; Thu, 30 Oct 2014 09:25:13 +0100 (CET)
**To** - =?utf-8?q?BBBNICK_=E2=99=A5_?= <bbb@seznam.cz>
**Subject** - Psycase
**Received** - from 94.112.234.2 (X-Forwarded-For: 94.112.234.2)
by mail1006.centrum.cz (centrum.cz multimail) with HTTP
**Date** - Thu, 30 Oct 2014 09:25:13 +0100
**From** - "AAANICK"<aaa@centrum.cz>
**X-Mailer** - Centrum Email 5.3
**X-Priority** - 3
**X-Original-From** - aaa@centrum.cz
**MIME-Version** - 1.0
**Message-Id** - <20141030092513.6739161F@centrum.cz>
**X-Maser** - Georgo
**Content-Type** - multipart/mixed;
boundary="_ca29_———-4e2457031f7e11db79f96b96a2ed0a93"

–_ca29_———-4e2457031f7e11db79f96b96a2ed0a93
**Content-Type** - multipart/alternative;
boundary="_d32c_———-af8a769a74a5832ba0463e8d5d76e9d4"

–_d32c_———-af8a769a74a5832ba0463e8d5d76e9d4
**Content-Type** - text/plain; charset=UTF-8; format=flowed
**Content-Transfer-Encoding** - 8bit

–_d32c_———-4249ab80bf3aa47d6637454d2d7fbe26
**Content-Type** - text/html; charset=UTF-8
**Content-Transfer-Encoding** - 8bit

–_d32c_———-4249ab80bf3aa47d6637454d2d7fbe26–

–_ca29_———-3d11c72652c93f53f08580f7069185ac–

---

Z hlavičky jsem získal poměrně mnoho specifických informací, které jsou pro běžného uživatele nic neříkající, ale pro někoho, kdo už třeba ví, co se dá zjistit z *IP adresy*, může být jednoduché, zjistit, odkud mu *email* přišel. Jak už jsem výše zmínil, tak se uživatel nedopátrá přesně na metr, odkud mu email přišel, ale v rámci přesného určení státu a města je šance poměrně vysoká. *Hlavičky emailu* jsou založeny na standardech, kterými se řídí služby, jež umožňují *emaily* odesílat. Přesto má každá služba svou specifickou hlavičku se svými položkami a z různých emailových serverů tedy budou mít hlavičky různé podoby.

### 2.3.4 WWW

*WWW* [15], *World Wide Web* nebo zkráceně web je označení pro aplikace, které využívají internetový *protokol HTTP* pro výměnu hypertextových dokumentů při komunikaci dvou zařízení. Pro zabezpečenou komunikaci se používá protokol *HTTPS*, který navíc celou komunikaci šifruje a zabraňuje tak odposlechu. Chce-li uživatel jít na stránku s adresou *www.vsb.cz* [6], tak tuto adresu zadá do prohlížeče, který se zadanou stránku (hypertextový dokument) pokusí vyhledat a pokud existuje, tak následně zobrazí výsledek. Na pozadí komunikace uživatel posílá dotaz s řadou informací na konkrétní server, který následně odpoví. Tato komunikace funguje na principu dotaz-odpověď. Pro dotaz existuje tzv. *hlavička dotazu (Request Headers)* a pro odpověď tzv. *hlavička odpovědi (Request Response)*, kde jsou tyto údaje posílány. Hlavičky jdou dnes snadno zjistit pomocí různých aplikací nebo je možné jejich zobrazení téměř v každém moderním prohlížeči *(Mozilla Firefox, Google Chrome)*. Jednotlivé dotazy na sobě nezávisí i přesto, že jsou následující dotazy na stejném serveru. Díky této vlastnosti je *protokol HTTP* nazýván bezstavový. Pro příklad komunikace uvedu, jak vypadají jednotlivé hlavičky v případě, kdy jsem chtěl zobrazit internetové stránky *www.vsb.cz* [6]:

**REQUEST HEADERS**

**Accept** - text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*-;q=0.8
**Accept-Encoding** - gzip, deflate, sdch
**Accept-Language** - en-US,en;q=0.8
**Connection** - keep-alive
**Cookie** - JSESSIONID=FBB44DA3DC3096A008FBA346B40481B8; __utmt=2;__utma=-247725952.1515662674.1415369926.1418128855.1418145181.21; __utmb=2477-25952.1.10.1418145181; __utmc=247725952; __utmz=247725952.1415811429.6-.2.utmcsr=google—utmccn=(organic)—utmcmd=organic—utmctr=(not%2-0provided)
**Host** - www.vsb.cz
**User-Agent** - Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.71 Safari/537.36

---

**RESPONSE HEADERS**

**Connection** - close
**Content-Length** - 0
**Content-Type** - text/html;charset=UTF-8
**Date** - Tue, 09 Dec 2014 17:13:17 GMT
**Location** - http://www.vsb.cz/cs/
**Server** - nginx/1.2.8

Položky v hlavičce se liší v závislosti na stránce, na kterou uživatel přistupuje. Z uvedeného příkladu jsou vidět nejčastější položky, které jsou součástí téměř každé komunikace. V tomto případě je odpověď serveru poměrně krátká a nijak nezajímavá. U složitějších

webových stránek může být hlavička odpovědi mnohem rozsáhlejší a může obsahovat mnohem více položek.

Pro uživatele je mnohem zajímavější *hlavička dotazu*, kde jsou vidět položky, které jsou posílány na server. Z těchto údajů lze vyčíst např. kódování prohlížeče na uživatelově straně, z čehož je možné odvodit např. stát, kde se uživatel nachází. Dále se server může dozvědět jaký internetový prohlížeč a operační systém návštěvník serveru má, a to z položky *User-Agent*.

Už jsem se zmínil o tom, že komunikace na internetu je bezstavová. Jak je potom možné, že po opakovaném příchodu na nějakou stránku, kde už byl uživatel přihlášený, je automaticky přihlášený, aniž by zadával přihlašovací údaje. K tomuto účelu vznikly *HTTP cookies* zkráceně *cookies*, což je malé množství dat, které server pošle prohlížeči a ten si je uloží do uživatelova zařízení. Při další návštěvě toho stejného serveru prohlížeč zjistí, zda má nějaké data z tohoto serveru uložené a pokud ano, tak je serveru pošle zpět, který je dále zpracuje a uživatele např. automaticky přihlásí. Díky těmto vlastnostem jsou servery schopné identifikovat uživatele nezávislé na jeho *IP adrese*, stačí, když bude používat stejné zařízení, kde jsou *cookies* ukládány. Dalším známým použitím této technologie je nákupní košík, kdy si uživatel přidává položky do košíku a po klikání na *e-shopu* položky v košíku zůstávají.

Ačkoli *HTTP cookies* ušetří mnoho psaní na klávesnici, tak sebou přináší určitá rizika, protože do počítače se ukládá řada dat ze serverů, které uživatel navštívil a servery tak můžou sledovat uživatelovy zájmy. Samotné data přitom nemusí ukládat autor stránky, ale může je ukládat klidně nějaký reklamní *banner*. Pro představu uvedu příklad, kdy uživatel vyhledává cokoliv o sportovních motorkách. Použije nejznámější český vyhledávač na stránkách *www.seznam.cz* [14]. Každé jeho vyhledávané slovo se uloží do *cookies*, až jich tam nakonec bude celá řada. Po čase uživatel opět přijde na tyto stránky a předtím, než se mu stránka načte, tak se v dotazovací hlavičce pošlou *cookies*, které uživatel má z minulosti uložené. Chytrý algoritmus na serveru zpracuje *cookies*, které říkají, co uživatel vyhledává a jeho výsledkem je, že uživatel se zajímá o sportovní motorky. Po tomto zjištění server na stránkách zobrazí cílené reklamy na koupi motorky nebo různé motorkářské *e-shopy*, od firem, které si tyto reklamy zaplatily.

Uživatelé si díky těmto reklamám mohou myslet, že je někdo sleduje a nabourává jejich soukromí. Ve skutečnosti jsou to programy, které toto provádí, ale uživatelovo soukromí je takto skutečně nabouráno, protože tyto cílené reklamy nejsou náhoda. Pokud se tyto informace navíc ukládají, tak si je autor této aplikace může klidně prohlédnout. Pro uživatele, kteří nechtějí být takto sledováni, tak je pro ně řešením v prohlížeči *cookies* vypnout. Prohlížeč pak žádné data ukládat nebude a nedojde k tomu, že uživateli vyskakují reklamy s produkty, o které se zajímá. Problém pak nastává, pokud servery vyžadují mít *cookies* zapnuté, jinak stránky nebudou fungovat. V takovém případě pak záleží na každém z nás.

## 2.4 Anonymizující techniky

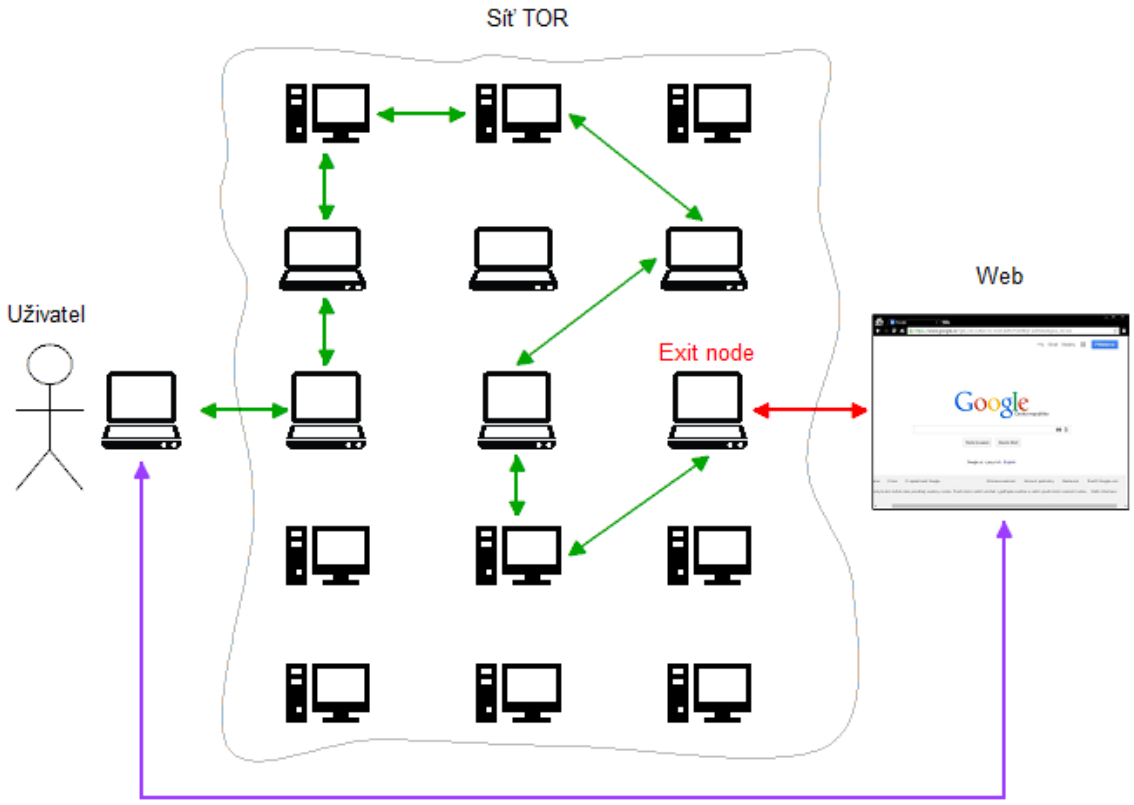
Být anonymní na internetu je obecně spojováno s pácháním nelegálních činností, jakými může být např. stahování nelegálního obsahu, posílání výhružných *emailů*, pornografie atd. V těchto případech dává použití anonymity smysl, ale drtivá většina uživatelů na internetu si neuvědomuje, že čím je uživatel anonymnější, tím více je také v bezpečí. Být anonymní dává smysl, i když uživatel nehodlá páchat nějakou trestnou činnost. Dosud jsem dokazoval, jak internet není anonymní a vyvrátil jsem tak mnoho domněnek o jeho anonymitě. Přes uvedené fakta přece jen existují způsoby, jak si anonymitu zachovat.

Důvody být anonmyní tedy existují, a tak vznikly techniky [16], které anonymitu dovolují a ty nejzajímavější z nich rozeberu v této kapitole. Dokonalá anonymita na internetu neexistuje, protože vždy existuje šance, že uživatel bude vystopován, ale at' už existuje jakákoliv šance odhalení pravé anonymity, tak ti nejlepší *crackeři (hackeři)*, kteří utočí na organizace po celém světě, svou anonymitu skrývají dokonale, protože dodnes jsou na ně tyto organizace ve většině případů krátké. Tyto organizace, mezi kterými jsou nejvíce zastoupené filmové společnosti, vedou proti útočníkům neustálý boj a ti většinou nikdy nejsou dopadeni.

### 2.4.1 TOR

Mezi nejkvalitnější techniky pro skrytí anonymity patří použití anonymizující sítě *TOR*, která zajišťuje anonymitu *IP adres* a její použití je zcela zdarma. Stačí si stáhnout klientskou část, nainstalovat a začít jí plně využívat. *TOR* neboli *The Onion Router* je přeloženo do češtiny jako *cibulové routování*.

*TOR* funguje na principu modelu *klient-server*, kdy se uživatel přes klientskou aplikaci připojí do sítě *TOR*, kterou bude následně využívat. Ihned po navázání spojení se vytvoří náhodná sada uzlů, přes které bude komunikace probíhat. Každý uzel v navázaném spojení ví pouze o uzlu předcházejícím a o žádném jiném. Síť *TOR* je tvořena dobrovolníky, kteří se dají podle jejich umístění v síti rozdělit na *vnitřní* a *vnější* uzly tzv. *exit nody*. Příklad takového spojení jsem znázornil na obrázku 2. Uživatel navázal spojení se síti *TOR* a jeho komunikace probíhá přes zeleně vyznačené šipky, až se dostane k *exit nodu*, přes který se dostane na požadovaný server. *IP adresa*, pod kterou se uživatel připojí k požadovanému serveru bude z *exit nodu* a nikoli z uživatelova. Většina uživatelů na internetu *TOR* nepoužívá a jsou spojeni přímo s daným serverem, jak je naznačeno fialovou šipkou.
*TOR* zajišťuje anonymitu, pokud se uživatel drží správných pokynů, které najde v manuálu služby. Teoreticky *TOR* není schopen 100% anonymitu zajistit. Posílaná data jsou sice už od uživatelova zařízení šifrována až po *exit node*, ovšem na výstupu *exit nodu* už je obsah komunikace vidět a pokud si uživatel nedá pozor, co o něm např. jeho internetový prohlížeč říká, tak je použití *TORu* celkem zbytečné. Proto je nutné dodržet přesné instrukce, jelikož vnějším uzlem může být kdokoliv a třeba i ten, kdo komunikaci vidět chce. Může se jednat např. o subjekty, které vlastní autorská práva různých děl nebo instituce, které chtějí monitorovat internet. Dalším problémem je, že mnoho serverů zná *IP adresy*, které pochází ze sítě *TOR*. Tyto *IP adresy* jsou na těchto serverech blokovány a




Obrázek 2: Ukázka spojení přes síť *TOR*

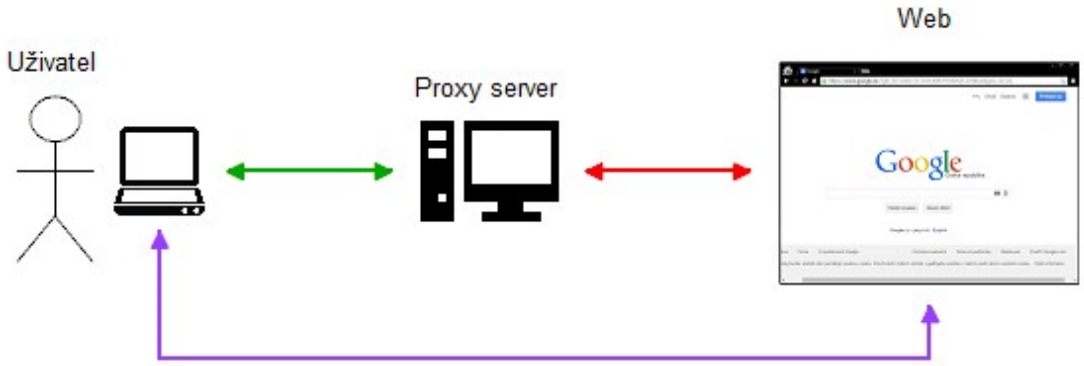
nikoho s touto *IP adresou* na svůj server nepustí, protože si tím tak zajistí větší ochranu, kdyby chtěl někdo náhodou porušovat zákon.

Pro skrytí anonymity na internetu je *TOR* poměrně dobrá volba, pokud uživatel nedělá věci v rozporu s legislativou dané země. Pro úplně zaručenou anonymitu pomocí *TORu* by bylo nutné mít kontrolu nad uživateli, kteří ho používají, ale to není možné zaručit. *TOR* už je navíc teď z určité části monitorován *Národní bezpečnostní agenturou (NSA)* [18]. A z pohledu uživatele, který neví, kdo se ukrývá na *exit nodu*, nemusí být úplně bezpečný. Dalším rizikem může být to, že uživatel používá *TOR* záměrně k páchání trestné činnosti a použitím *TORu* je tak skrytý daleko za *exit nodem*, který patří právě uživateli, jež o tomto nemá nejmenší zdání. Velký problém se zákonem může mít právě tento uživatel, který o ničem nevěděl. Používání *TORu* není úplně rychlé, protože komunikace probíhá přes několik spojovacích uzlů (zařízení) a používat ho pro stahování není zrovna nejlepší nápad. Zpomalení už logicky vyplývá z toho, že mezi uživatele a cílový web je vložen nějaký prostředník v podobě sítě *TOR*.

### 2.4.2 Proxy servery

Další alternativou k *TORu* můžou být pak tzv. *anonymizující proxy servery*, kterých je celá řada a které fungují podobně jako *TOR*. Rozdíl je v tom, že mezi uživatele a server, na který se chce uživatel dostat, je umístěn *proxy server* a přes něj se pak uživatel připojuje na cílové servery. *Proxy serverů* může být více v řadě, ale čím více jich bude, tím bude komunikace pomalejší. Vzhledem k vytížení nejsou tyto servery nejrychlejší, ale pro normální použití jako je tomu u *TORu*, se využít dají. Oproti *TORu* jsou zde větší rizika monitoringu, protože spojení probíhá většinou přes jediný server, který může provozovat kdokoliv a třeba i ten, kdo někoho sledovat chce. Mezi nejznámější *proxy servery* patří: *Proxify, Ninja Clock, AnonyMouse, AnonyMizer, JAP*.

Princip komunikace *proxy serverů* jsem zobrazil na obrázku 3, kde je vidět pouze jeden prostředník mezi uživatelem a serverem, kterým je právě *proxy server*. Uživatel přistupuje na *proxy server* se svou *IP adresou*, což je naznačeno zelenou šipkou. Následně komunikuje *proxy server* přímo s požadovaným serverem, což je naznačeno červenou šipkou, ale už s jinou *IP adresou*, než s původní uživatelovou. Cílový server nekomunikuje s uživateli, ale s *proxy serverem*. Komunikace je za normálních podmínek naznačena opět fialovou šipkou. Problémy s *proxy serverem* jsou podobné jako se sítí *TOR*. Servery mohou adresy vycházející z *proxy serveru* blokovat a navíc u *proxy serveru* není anonymizace tak důkladná, jako je tomu u *TORu*. Uvnitř *TORu* je komunikace šifrována a přímý obsah této komunikace může být hodně problematické odhalit.




Obrázek 3: Ukázka spojení přes proxy síť

### 2.4.3 VPN Servery

V současné době jsou na internetu rozmnoženy také *VPN servery*, které rovněž nabízí anonymitu za pomocí *VPN sítí*. Technika je podobná jako u *proxy serverů*. Uživatel se připojí přes *VPN klienta* na server a dále pak vystupuje pod *IP adresou* vycházející z *VPN sítě* místo své *IP adresy*. Hlavním rozdílem oproti *proxy serverům* je, že *VPN servery* komunikaci šifrují a chrání tak uživatelova data, kdežto *proxy servery* jsou jenom prostředníkem

při komunikaci. *Proxy servery* umožňují komunikaci rychlejší, protože u nich odpadá šifrování, ale samozřejmě to vždy platit nemusí.

*VPN sítě* se používají většinou v korporacích pro privátní sítě, kde chtějí mít zaručené, že danou síť využívá ověřená osoba. Ve všech případech poskytovatelé anonymních služeb zaručují 100% anonymitu, ovšem zase taková jistota to není a vždy stojí za úvahu, zda uživatel bude věřit něčemu, co je napsané na stránkách dané služby pocházející z úplně jiného kontinentu nebo bude věřit svému selskému rozumu. Takovým poskytovatelem může být totiž opět někdo, kdo službu poskytuje za účelem monitorování internetu a já bych s důvěrou k těmto službám byl opatrný. Mezi nejznámější poskytovatelé anonymních *VPN klientů* lze zařadit: *HideMyAss, UltraSurf, proXPN, tunnelBear VPN, CyberGhost VPN*.

### 2.4.4 P2P Sítě

*Peer-to-peer* sítě jsou počítačové sítě, kde probíhá symetrická komunikace mezi počítači, kde každý z nich je schopen iniciovat nebo vykonat požadované operace. Společným znakem těchto síti je sdílení souborů, či textových zpráv na kterémkoliv počítači, který je součástí této sítě. S postupem času se z požadavků těchto sítí slevuje a někdy se stává, že po vypadnutí konkrétního počítače není možná jeho náhrada. Pro uživatele je zde většinou problém, že při stahování souborů jsou soubory zároveň nabízeny. To je problém v mnoha zemích, kde se tímto porušuje tamní legislativa.

Z pohledu skrytí anonymity, tedy skrytí *IP adres*, vylepšují autoři aplikací postavených na *peer-to-peer* provoz o šifrování, kdy je anonymity dosaženo pomocí tzv. *překryvných sítí*, kde uživatelova *IP adresa* je označena pseudonymem. Tento pseudonym je odvozený od uživatelova veřejného klíče a zároveň slouží jako jeho adresa. To umožňuje posílat konkrétní soubory nebo textové zprávy tomuto uživateli a přitom skrývá jeho *IP adresu*. Síť, která takto fyzicky skrývá uživatele pomocí *překryvných adres*, se nazývá *překryvná síť*. Pro tyto sítě je potřeba mít nainstalovaného klienta, který vykonává za uživatele veškerou důležitou práci. Mezi takové peer-to-peer sítě např. patří: *Gnutella, FastTrack, eDonkey, BitTorrent*.

### 2.4.5 I2P Sítě

Technika *I2P sítí* neboli *Invisible internet project* založený na překrývání sítí je velice podobná *cibulovému routování* jako u *TORu*, protože neposkytuje žádné konkrétní služby a všechna přenášená data jsou zde navíc několikrát šifrována. Zde se jedná pouze o nástavbu nad protokolem *TCP/IP*, který má zaručit anonymitu. Tato síť je navržena tak, aby bylo velice snadné do ní vložit jakoukoliv službu, jakými může být např. sdílení souborů. Distribuce sítě je prováděna dynamicky a žádná strana není považována za důvěryhodnou.

# 3 Digitální stopa

*Digitální stopu* lze popsat jako jakékoliv data, které vzniknou používáním digitálních služeb a zůstanou někde uložena bez ohledu na to, zda uživatel o uložení ví nebo ne. V dnešní době je digitální stopa spjata především s používáním internetu, ale ve skutečnosti je spojena s každým zařízením, které lze použít pro připojení k internetu. Pro mnohé uživatele se internet jeví jako virtuální svět zcela anonymní, ovšem ve skutečnosti je tomu úplně jinak. Každý uživatel, který kdy internet použil, za sebou zanechal mnoho informací, které se považují za tzv. *digitální stopu*. Tato *digitální stopa* může být viditelná pro ostatní uživatele internetu nebo ji vidí pouze zasvěcení, a to mohou být např. administrátoři serverů, které uživatel navštívil nebo *provider*, který danému uživateli poskytuje internetové připojení. V neposlední řadě je může vidět nějaký útočník *(cracker)*, který se k nim dostane díky svým dokonalým znalostem. Je třeba mít taky na paměti, že na internetu není anonymní nikdo, protože pokud na něm uživatel je, tak má přidělenou *IP adresu* a *provider* zaznamenává, kdo a kdy danou *IP adresu* použil. Podle času a *IP adresy* pak lze celkem snadno daného uživatele vypátrat. Tyto informace *provideři* musí ze zákona archivovat několik let.

*Digitální stopa* může mít mnoho podob. Mezi nejrozšířenější *digitální stopy* na internetu viditelné pro ostatní uživatele patří komentáře, jakákoliv aktivita na sociálních sítích, komentáře pod různými články, jakákoliv diskuze na fórech, vlastní internetové stránky, blogy, registrace, členství na internetových fórech a mnoho dalších. S trochou štěstí můžeme podle jednoho komentáře na jakékoliv internetové diskuzi zjistit, že autor příspěvku je ženatý muž žijící v Opavě, který má dvě děti, mobil značky Apple, auto značky BMW, pracuje jako finanční poradce a jeho názory jsou silně pravicové. Pokud se dá toto všechno zjistit z jednoho příspěvku, tak je to určitě muž, který není v prostředí internetu moc ohleduplný, což může způsobit to, že kdekdo si o něm může cokoliv zjistit a nabourat tak jeho soukromí. I když mu to nemusí vadit, že kdokoliv si o něm může zjistit i soukromé záležitosti, tak můžou být tyto informace ve výsledku proti němu použity. Případného útočníka zjištění takových informací téměř nic nestojí.

Za neviditelné *digitální stopy* na internetu lze považovat např. *log soubory* na serverech, které monitorují uživatelův pohyb na daném serveru. *Email* obsahuje taky mnoho informací, jak jsem uvedl v kapitole 2.3.3, o kterých běžný uživatel neví, protože pro něho nejsou důležité a zajímá se především o odesílatele, předmět, obsah *emailu* a příjemce. Ve skutečnosti každý příjemce *emailu* si může zobrazit *hlavičku emailu*, která obsahuje informace, kterým rozumí většinou jen člověk z oblasti *IT*, proto je většinou hlavička skryta a jsou zobrazeny pouze informace spjaté s funkcí *emailu* (odesílatel, předmět, obsah *emailu*).

*Digitální stopa* není spjata jen s internetem, ale lze se s ní setkat i v *off-line* prostředí, kde se skrývá např. v souborech v podobě tzv. *metadat*, což jsou strukturovaná data popisující data. K vytvoření *metadat* pochopitelně není internet potřeba, protože se vytvoří pouhým vytvořením konkrétního druhu souboru, u kterého je možné uchování specifických informací. Například se stačí podívat na detailní informace obrázku a můžeme zjistit, kdy a kde byl pořízen. Zvídavý manžel tyto detailní informace prozkoumá a nečekaně zjistí, že jeho manželka se před týdnem fotila u kamarádky s přáteli a ne včera, jak ho původně informovala. Tak kde jeho manželka včera potom byla?

## 3.1 Klasifikace

*Digitální stopy* lze rozdělit do dvou hlavních kategorií podle způsobu, jak byly vytvořeny z uživatelského pohledu. Pokud ji uživatel vytvořil vědomě, tak se jedná o *digitální stopu aktivní*. Je-li *digitální stopa* vytvořena na pozadí digitálních služeb, kde uživatel nevidí žádnou informaci o jejím vytvoření nebo jednoduše k ní nemá přístup, tak se pak jedná o *pasivní digitální stopu*.

1. **Aktivní** - mezi *aktivní digitální stopy* se řadí takové stopy, o kterých daný uživatel ví a vytváří je záměrně za účelem sdílení informací o své osobě pomocí sociálních médií a internetových stránek. Uživatel rovněž může např. fotografii doplnit *digitální stopou* v podobě datumu nebo místa vytvoření, pokud se to neděje automaticky. Tato *digitální stopa* jde dále rozdělit v závislosti na prostředí, ve kterém byla vytvořena.

    - **On-line prostředí (internet)**
        - Jakákoliv veřejně viditelná aktivita (např. komentáře).
        - Jakékoliv sdílení informací (např. profilové informace na fórech).
        - Publikace internetových stránek.
        - Registrace na fórech (určují uživatelovy zájmy).
        - Sociální sítě, *chat*, *email*, atd. ...
    - **Off-line prostředí**
        - Jakákoliv data uložena v jakémkoliv souboru (např. textové soubory).
        - Různá metadata souborů, o kterých uživatel ví, atd. ...

2. **Pasivní** - *pasivní digitální stopou* lze nazvat všechna data, která byla získaná bez uživatelova vědomí či souhlasu. Většinou to jsou data, která mají uživatele poškodit nebo ho monitorovat.

    - **On-line prostředí (internet)**
        - *Keylogger* (Program, který zachytává všechny kliknutí na klávesnici a myši. Na internetových stránkách by to bylo možné vyřešit *javascriptem)*.
        - *Server logy* (Servery si na pozadí můžou uchovávat řadu dat, o kterých se nikdy nedozvíme. Minimálně se může jednat o data, které jsem rozebral v kapitole 2.3.4. V mnoha případech se mohou ukládat uživatelovy akce. K těmto datům pak mají přístup pouze administrátoři serveru nebo jsou automaticky zpracovávaná).
        - *HTTP cookies* (Popsány v 2.3.4),
        - Internetové chyby (Tzv. *web bugy*, díky kterým jsou zneužívány špatně naprogramované služby jako *email* či samotné internetové stránky.)
        - Historie internetového prohlížeče (autoři prohlížečů si pro sebe mohou ukládat zajímavá data a pak je na pozadí kdykoliv odeslat na požadovaný server, kde budou dále zpracována).

- **Off-line prostředí**
  - *Keylogger* umístěn v počítači, aniž by o něm uživatel věděl (např. k získaní hesla)
  - Soubory přístupné pouze administrátorům (většinou logy a uživatel nikdy nezjistí, co se loguje)

## 3.2 Zneužití

Na internetu se pohybuje mnoho podvodníků a zlodějů stejně tak, jako se pohybují v reálném životě. Díky internetu nemusí dojít ani k fyzickému kontaktu mezi oběma stranami a přesto může podvodník slavit úspěch, protože obět'se chovala na internetu velmi neopatrně. Proto je potřeba se na internetu chovat obezřetně a vyvarovat se případných nepříjemností, jelikož v dnešní době ani tam není nikdo v bezpečí, tak jako všude jinde a nepomůže tomu ani fakt, že se jedná o trestnou činnost[20]. Elektronická data jsou navíc levná, dobře zpracovatelná a snadno přístupná.

Pohybem na internetu za sebou uživatel zanechává informace a to je to, co patří mezi nejdůležitější fakta při zneužití *digitální stopy*. Použitím chytrých vyhledávačů a kombinací vyhledaných údajů lze zjistit o uživateli mnoho informací včetně jeho vzhledu, názorů, přátel aj. Důsledek generování *digitálních stop* je velmi zákeřný, protože uživatel může nevědomky ztratit soukromí a např. při získávání práce se uživatelovo zveřejňování informací může obrátit proti němu, protože personalisté různých firem si takto ověřují případné adepty o zaměstnání [3].

Další nepříjemností se pro uživatele může nastat, pokud podvodník (zloděj identity) nashromáždí o uživateli tolik informací, že se za něj může vydávat, aniž by o tom uživatel věděl. V případě, že podvodník zná uživatele i osobně, tak může všechny informace spojit dohromady a velmi mu tak znepříjemnit život např. v podobě pomluvy, okradení nebo nabourání se do jeho soukromých záležitostí. Samozřejmě uvedené příklady zneužití jsou v rozporu s legislativou většiny států na světě a řešení již existujícího problému určitě bude trvat dlouho, přičemž pověst poškozeného se napravuje jen velmi těžko. Ačkoliv se nyní uživateli může zdát, že na internetu lze všechno, tak se mýlí. Internet své hranice má a tím je právo.

### 3.2.1 Ztráta soukromí

Soukromí je nedílnou součástí každého z nás a ve většině států je soukromí chráněno tamními zákony. Do soukromí spadá především oblast o ochraně osobních údajů a ochrana těla týkající se intimních oblastí. Nikdo tedy nemá právo o nikom shromažďovat informace a jakkoliv je používat, ale tomu jen tak nikdo nezabrání. Nezodpovědným chováním na internetu navíc uživatelé sami ke ztrátě soukromí přímo vybízejí.

### 3.2.2 Krádež identity

Poskytne-li na internetu uživatel dostatek informací, ze kterých je schopen útočník sestavit něčí identitu, tak tady existuje možnost, že se za tohoto uživatele bude útočník

vydávat. Uživateli tak může způsobit celkem velké nepříjemnosti, protože zná-li útočník jeho jméno, příjmení, přezdívku nebo *email*, tak může vytvářet uživatelovy profily na sociálních sítích a ne jenom tam. Tímto útočník získá cizí přátele a díky nim se může dozvědět informace, které by jemu normálně neposkytli.

V jiném případě se nemusí jednat o vytváření falešných profilů na různých internetových stránkách a diskuzních fórech, ale útočník může uživatele poškodit pomluvou. Buď ho může pomluvit jako zcela vymyšlená osoba nebo při získání většího množství informací se za něj může útočník vydávat (ukradne mu identitu) a pod uživatelovým jménem bude rozšiřovat informace poškozující uživatelovo jméno nebo dokonce jméno někoho jiného. Vinu pak nese poškozený uživatel a případné očištění jména není vůbec jednoduché.

Krádež identity může být taky zneužita, pokud chce útočník získat přihlašovací údaje k různým internetovým účtům. Má-li dostatek informací o uživateli, může kontaktovat administrátory příslušných stránek, že ztratil přihlašovací údaje a pokud bude administrátor důvěřivý, tak je může útočníkovi poskytnout. Ovšem pokud administrátor důvěřivý nebude a bude požadovat i detailní informace o uživateli, které by ho více identifikovaly, tak útočníkem může být pak třeba uživatelův kamarád, který tyto informace znát bude a administrátor mu požadované informace nakonec stejně poskytne. Taková krádež identity by se neměla stát na internetových stránkách s velice citlivými údaji, jakými je internetové bankovnictví apod. Tam už ale dochází k mnoha ověřením, které už nejsou snadno zneužitelné. Dále uvedu pár příkladů, kde může na internetu docházet k získávání informací, které jsou zneužívány při poškozování osob.

- **Sociální sítě**
  Fenoménem dnešní doby jsou sociální sítě. Mezi nejznámější patří např. *Facebook [21], Google+* [22] nebo *Twitter* [23]. Uživateli nabízí mnoho pozitiv a hlavně snadnou komunikaci s přáteli, i když žije na druhém konci světa. Pozadu není ani snadné sdílení fotek s přáteli, hraní různých her aj. možnosti zábavy. Přes všechny tyto jinak užitečné možnosti je tu jedna negativní vlastnost, kterou je poskytování informací. Jednak je uživatel poskytuje sociální síti a pokud si nedá pozor, tak jsou informace poskytnuty úplně každému, tak jak je to psáno v následujícím příkladu o podmínkách používání *Facebooku* ohledně sdílení informací. *Pokud publikujete obsah nebo informace s použitím nastavení Veřejné, znamená to, že povolujete všem (včetně osob mimo službu Facebook) přístup k těmto informacím, jejich použití a jejich spojení s vámi (tj. s vaším jménem a profilovou fotkou)* [24]. Sice jsem nezjistil, kolik profilů na zmiňovaném *Facebooku* je veřejných, ale po krátkém vyzkoušení několika náhodných jmen si troufám napsat, že jich bude víc jak 50%. Vzhledem k tomu jaké problémy mohou nastat díky těmto rizikům, tak uživatelů, kteří si uvědomují riziko veřejného profilu, je z globálního hlediska velmi málo.
  Pro objasnění problematiky jsem si vybral již zmíněnou sociální síť *Facebook* a dále se bude vše vztahovat k této sociální síti, protože u všech ostatních sítí je to obdobné nebo dokonce úplně stejné. Samotný problém nastává už při registraci, jelikož pro registraci je potřeba potvrdit přečtení licenčních podmínek, se kterými uživatel musí souhlasit, jinak se nezaregistruje. Přečtení těchto základních podmínek mi tr-

valo okolo 30 minut a to nepočítám další ustanovení, která se skrývají pod odkazy, které jsou v těchto podmínkách uvedeny. Navíc některé informace jsem si musel přečíst vícekrát, jelikož právnicky napsaný text je trochu komplikovanější než normální, ale to je spíše individuální záležitost. Kdo tyto podmínky ale přečetl, než se zaregistroval? Silně pochybuji, že každý uživatel si je prošel, vždyť už jenom samotný rozsah odradí od jejich přečtení. Většina si asi řekne, *Facebook* má přece každý, tak proč by se zdlouhavým čtením zabývat?
Prostředí *Facebooku* působí na uživatele velice přátelsky a důsledkem toho pak můžou být uživatelé méně obezřetní, můžou zveřejňovat své fotky, pouští se do různých diskuzí a to nejhorší, zveřejňovat o sobě citlivé informace, které mohou být dále zneužívány. Navíc je to ideální prostředí pro šíření *malware*. Osobně si nemyslím, že by *Facebook* měl znát veškeré mé soukromé údaje, i když zaručuje, že nebudou nikdy zneužity. Proč tyto informace potom *Facebook* chce a brání se jakémukoliv smazání čehokoliv, co uživatel už na něj vložil? Při mizerném nastavení viditelnosti účtu může soukromé informace vidět dokonce kdokoliv a podvodníci se mohou v kyberprostoru pohybovat stejně tak, jako v reálném světě, tak by si to každý měl uvědomit.

- **Internetové diskuze/fóra**
  Podobně jako na sociálních sítích, tak i na ostatních internetových stránkách existuje možnost sdílet informace v podobě různých komentářů pod články, *chatů* nebo probíráním různých témat na diskuzních fórech. Vložením jakéhokoliv komentáře uživatel poskytuje informaci s ním související a většinou je možnost vložení komentáře podmíněna registrací na příslušné stránce nebo je přinejmenším požadováno vyplnění jména při vkládání. Tyto vyplněné informace o uživateli jsou většinou viditelné všem a tak není problém zjistit, jak se jmenuje autor příspěvku nebo jaký má *email*. Existují sice uživatelé, kteří si jména a *emaily* vymýšlejí, ale pokud jejich chování na dané stránce není korektní nebo pokud hýří vulgarismy, tak dostane zákaz v podobě zablokování profilu nebo zablokování *IP adresy*, přičemž se pak už na konkrétní server útočník nedostane.
  Dalším problémem diskuzních fór, různých sdružení, či jakýchkoliv podobných internetových stránek je zobrazování jednotlivých členů na daných stránkách. U těchto členů bývají velice často nejrůznější informace závisející na konkrétním webu, které dále konkrétněji popisují jednotlivé uživatele. Například na stránkách věnující se fanouškům aut značky *Mercedes*, se zobrazení informace, že nějaký uživatel vlastní nejnovější model v ceně několika miliónů, může zpočátku jevit nevinně, ale tento uživatel se stává potenciální oběti, protože na daných stránkách se můžou pohybovat zloději aut, o kterých nemá nikdo nejmenší zdání, protože uživatelé nikdo nekontroluje. Ti si pak můžou s pomocí internetových vyhledávačů dále o tomto uživateli zjistit informace o jeho bydlišti. Pokud majitel *Mercedesu* bydlí v menší obci a tuto informaci má zveřejněnou, dále tam má vložené fotky ze kterých je možné identifikovat jeho bydliště, tak pro zloděje není problém použít mapy na internetu a s trochou štěstí najde bydliště celkem rychle. Každý si pak dovede představit, co bude následovat.

- **Profily**
  Mnoho internetových stránek a nejrůznějších diskuzních fór všeho druhu požaduje pro své využívání registraci, tak jako např. *Facebook*. Při zakládání profilu nebo po prvním přihlášení je po uživateli mnohdy požadováno, aby vyplnil svůj profil s nejrůznějšími osobními údaji. Velice často se stává, že tyto údaje jsou zcela irelevantní vůči obsahu daných internetových stránek. Nevidím jediný důvod, proč by diskuzní fórum zabývající se pěstováním rostlin mělo po uživateli požadovat vyplnění bydliště nebo mít jeho profilovou fotku, podle které by ho bylo možné identifikovat.

- **Vlastní blogy/internetové stránky**
  Pokročilejší uživatelé na internetu si už dokážou vytvořit vlastní internetové stránky a na nich zveřejňují různé osobní informace. Velice často se jedná o stránky s tématikou, o kterou se uživatel zajímá a tím o sobě nepřímo zveřejňuje informace. Pro méně pokročilejší uživatele jsou to tzv. blogy, na kterých si každý uživatel může velice snadno vytvořit stránky, pomocí jednoduchých technik. Ti pokročilejší si stránky ve stejném duchu naprogramují. Na těchto stránkách a blozích dochází opět ke zveřejňování informací.

## 3.3 Příklady použití/zneužití

Některé příklady použití či zneužití *digitálních stop* jsem už uvedl v předešlých kategoriích, ale zde uvedu konkrétní příklady, které jsou zajímavé z mnoha hledisek a díky kterým se stojí nad *digitální stopou* více zamyslet.

### 3.3.1 Please Rob Me

Projekt *Please Rob Me* [25] vytvořeny partou lidí, kteří se zabývají bezpečností na internetu, demonstruje, jak lehké je zneužít na první pohled nevinné informace týkající se určení polohy uživatelů pomocí internetové aplikace *Forsquare* [26] a sociální sítě *Twitter* [23]. Pomocí internetové aplikace *Forsquare* uživatelé zveřejňují informace o místech, kde se nacházeli nebo kde se právě nacházejí. Tohoto využila aplikace *Please Rob Me* tak, že tyto informace na *Forsquare* dala dohromady s uživatelovým jménem a bydlištěm a poté je zveřejnila ve smyslu: „Právě jsem na oslavě narozenin v restauraci na druhém konci města (= nejsem doma), tak mi můžete vykrást byt."

Na první pohled se můžou informace na *Forsquare* jevit celkem nevinně, ale projekt *Please Rob Me* jednoduchým příkladem tuto domněnku vyvrátil. Cílem tohoto projektu nebylo vykrádat byty a domy uživatelů *Forsquare*, ale poukázat na nebezpečí, které může na internetu nastat sdílením citlivějších informací. *Forsquare* na tento projekt zareagovali prohlášením, ve kterém sdělují, že poselství *Please Rob Me* pochopili a osobní informace uživatelů berou vážně, přičemž pracují na vylepšeních [27].

*Please Rob Me* nakonec službu zastavil, ale nebezpečí tím nebylo zažehnáno, jen bylo na něho poukázáno, protože stálé hodně uživatelů zveřejňuje na sociálních sítích informace o jejich aktuální poloze a pokud má zloděj k těmto informacím přístup, tak je problém na světě. Je třeba myslet na to, že zlodějem může být i uživatelův nejbližší kamarád.

### 3.3.2 Robin Sage

Další skvělý experiment [28], poukazující na hloupost uživatelů pohybujících se na internetu, představil americký bezpečnostní specialista *Thomas Ryan*. Na nejrůznějších sociálních sítích vytvořil fiktivní profil 25-leté atraktivní dívky, kterou pojmenoval *Robin Sage*. Toto jméno převzal z kurzů cvičení speciálních jednotek armády Spojených států, kde představuje určitou etapu. Velmi rafinovaným tahem bylo zvolení profilové fotky *Robin Sage* 4, která byla převzata z porno stránek a tak nebylo pochyb o atraktivitě této dívky. K tomu všemu jí byl vytvořen dokonalý životopis, ve kterém stálo, že je absolventka špičkové *IT* univerzity, má 10 let praxe a pracuje jako kyberanalytik počítačových hrozeb v jedné organizaci.

Po vytvoření profilu začal *Thomas* posílat žádosti o přátelství lidem především z oblasti bezpečnosti, armády a zpravodajství. Po době zhruba 2 měsíců získala *Robin (Thomas)* řádově stovky přátel, na což mnoho z nich poslalo *Robin* pracovní nabídky, pobídky na korektury vědeckých prací, ale také pozvání na různé schůzky. *Thomas Ryan* takto získal mnoho interních informací různých společností, mezi kterými byl např. i *Google*. Svým posudkem mohl zmařit nebo navést jiným směrem i nějakou vědeckou práci. Navíc získal *GPS souřadnice* amerických základen v Afghánistánu, protože její přítel ze sociální sítě tam operoval jako voják a poslal *Robin* své fotky, ve kterých byly ukryté *GPS souřadnice*, které do fotek ukládají modernější zařízení .

I v tomto případě bylo poukázáno na lidskou nemístnou důvěřivost a hloupost. Z velké ledabylosti a neověření si s kým konkrétní hříšníci komunikují, utekly poměrně citlivé informace. Kdyby byl *Thomas Ryan* nějaký *hacker* nebo terorista, mohl získané informace poměrně hodně zneužít. Jeho cílem ale nebylo někoho zneužít a poškodit, ale poukázat jaké rizika hrozí neopatrným zacházením s citlivými informacemi, protože zneužitím třetích stran může dojít i k ohrožení národní bezpečnosti.

Obrázek 4: Fotografie fiktivní dívky *Robin Sage* [29]

### 3.3.3 Radius

Fakt, že *digitální stopa* se dá zneužít, jak jsem zmínil v předchozích příkladech, je záporná stránka věci. Nyní se zmíním o jednom příkladu, jež využívá *digitální stopu* pozitivně a tím příkladem bude společnost *Radius*, což je softwarová společnost, která vznikla přejmenováním společnosti s původním názvem *Fwix*, založené *Darianem Shirazim*.

*Darian* začal rozvíjet svou kariéru v 15-ti letech, když začal obchodovat s počítačovým příslušenstvím. Zjistil, že zboží je nejlevněji vyráběno v Asii, a tak zkontaktoval asijské výrobce tohoto zboží, aby ho posílali přímo jemu. *Darian* vše pak prodával na aukční síni *eBay* [30] za vyšší cenu než pořídil. Tímto prodejem měl nečekaně velký obrat, čehož si lidé na *eBay* všimli a *Dariana* u sebe zaměstnali. Netrvalo dlouho a *Darian* pracoval ve společnosti *Facebook*, ale jelikož měl vyšší cíle, tak po dvou letech odešel a začal pracovat na svém projektu *Fwix*, který od něho chtěl koupit *Google*, ale ani s enormní částkou 3/4 miliardy [31] Google neuspěl.

*Fwix* shromažďoval informace ze sociálních sítích *(Facebook, Twitter, Forsquare, aj.)* a jiných veřejně dostupných zdrojů. Na základě získaných informací pak *Darian* dokázal zjistit uživatelův životní styl. Podle sestaveného profilu následně bylo možné spojit uživatele např. s produktem, který by si mohl koupit. Je-li podle sestaveného profilu uživatel velmi movitý a jeho zálibou jsou rychlé auta, tak *Darian* dokáže vyhodnotit míru pravděpodobnosti, s jakou si uživatel koupí nejnovější a nejrychlejší auto na světě, které právě přišlo na trh.

S těmito informacemi už *Darianův* systém nepracuje, ale prodává je firmám, kterých je okolo 27 miliónů. *Darianova* společnost se stala víc profesionální s řadou změn a dostala nový název *Radius*. Dnes vyvinutý systém pomáhá začínajícím podnikatelům nebo firmám lépe určit marketingové cíle a jejich úspěšný start od budoucna.

## 3.4 Prevence

Každý, kdo používá internet by si měl vyzkoušet, co vše je možné o něm dohledat a následně zvážit, zda-li jsou nalezené informace bezpečné nebo pro daného uživatele určitým způsobem nějak kompromitující. Pro vyhledávání těchto informací by měl uživatel použít více internetových vyhledávačů a postupně v každém z nich zkusit zadat své jméno, příjmení, *email*, přezdívky, login, telefonní číslo a adresu. Zjištění může být velmi znepokojivé a případná náprava už bude prakticky nemožná, protože i při smazání všech příspěvků, zrušení nebo deaktivace všech účtů zůstávají tyto data stejně uložena z různých archivačních důvodů, jen budou zneviditelněna, jak je tomu v případě u *Facebooku*.

Vzhledem k rizikům, které *digitální stopa* představuje, vznikly i nejrůznější metody, jak se tomuto na internetu bránit. Nejúčinnější metodou by sice bylo internet vůbec nepoužívat a pak by se nikdo nemusel ničeho bát, ale život bez internetu si dnes lze jen těžko představit. Pominu-li tuto možnost, tak největším nebezpečím každého uživatele jen on sám, protože *digitální stopa* je zneužívána právě díky selhání lidského faktoru, jakým je např. uživatelova nevědomost či lehkomyslné jednání v prostředí internetu. Bude-li se uživatel na internetu pohybovat a nevloží nikde ani písmenko, tak z hlediska *digitální stopy* mu nehrozí téměř žádné nebezpečí, ale je třeba mít na paměti, že uživatelův pohyb

není zcela anonymní, protože internet je postavený na *IP adresách*, jak bylo vysvětleno v kapitole 2.1. Navíc pokud už uživatel někde něco vloží nebo okomentuje, tak to prakticky nelze vzít zpět, protože internet se různě zálohuje a poskytovatelé internetových služeb se brání jakémukoliv smazání. Internet je pouze anonymní do takové míry, do které mu to povolí uživatelův *provider*. V následujících kapitolách uvedu ochranné metody před zneužitím *digitálních stop*.

### 3.4.1 Skrytí anonymity

Tato oblast je detailně popsána v kapitole 2.4. Aktuálně nejpoužívanějším prostředkem pro skrytí anonymity je *TOR*, ale služeb tohoto druhu neustále přibývá, tak v budoucnu možná dojde ke změně. Rizikem zde je, že uživatel neví, kdo službu provozuje. Anonymizující techniku je pro jejich plnou funkčnost nutno používat správně.

### 3.4.2 Nezveřejňování citlivých a osobních dat

Tento problém se týká především sociálních sítí a všech stránek, kde je možnost vkládání různých příspěvků nebo vytváření profilů. Uživatelé si často neuvědomují, že k těmto informacím nemají přístup pouze oni sami a že jednou vložené informace už většinou nejdou vzít zpět. Je zbytečné, aby uživatel zveřejňoval vše, co po něm daná služba chce a pokud není jiná možnost, tak teoreticky připadá v úvahu vymyslet si o sobě nepravdivé údaje. Osobně to takto praktikuji a nikdy si nikdo nestěžoval. V případě vyplňování informací, by se měl každý řídit pravidlem, že cokoli by neřekl cizímu kolemjdoucímu, tak by neměl ani na internetu zveřejňovat a to samé platí i o fotografiích. Uživatel by si měl dobře rozmyslet, jestli je důležitější např. se někomu pochlubit a mít vykradený byt nebo se raději nechlubit vůbec a pokud, tak jenom v rodinném kruhu. V této souvislosti bych ještě chtěl zdůraznit přidávání si přátel a to hlavně na sociálních sítích. Mnohdy má totiž jeden uživatel tisíce osob v přátelích a osobně zná sotva půlku. Tu druhou tvoří přátelé jen z internetu, o kterých nemá uživatel tušení, kdo vlastně jsou. Jeho představa se může odvíjet maximálně z fotografií, které můžou být podvrh a nový přítel není kamarád kamaráda, ale je to sériový vrah.

U služeb, kde je nutné dbát na zvýšenou bezpečnost, jako je např. internetové bankovnictví, se o poskytování a zveřejňování osobních informací uživatelé bát nemusí. Tyto služby jsou velmi privátní a mají k nim přístup pouze zaměstnanci dané korporace a samotný uživatel (zákazník). Přihlašovací údaje u těchto služeb jsou pro větší bezpečnost poskytovány pouze při osobním kontaktu a při ověřování identity zákazníka.

Rizikem u těchto služeb je, pokud jsou přihlašovací údaje poskytnuty třetí straně (podvodníkům). Podvodníci pro zisk přihlašovacích údajů používají metodu nazývanou *phishing*, česky rybaření. Ta spočívá v rozeslání podvodných *emailů*, kde jejich obsah žádá uživatele, aby si na uvedené stránce změnil heslo např. k internetovému bankovnictví. Uvedený odkaz na stránku je ale podvrh a vložené údaje jsou pak útočníkem lehce zneužitelné na skutečných stránkách dané banky. I když banky neustále před tímto rizikem varují, tak někdo z řady nezkušených uživatelů čas od času stejně naletí.

### 3.4.3 Více přihlašovacích jmen a hesel

Velmi problematickým faktem na internetu je, že uživatelé často používají ke všem službám stejné přihlašovací jméno a heslo. Pokud se stane, že tyto údaje získá útočník, tak má uživatel vážný problém. Stačí, když se uživatel zaregistruje na webových stránkách, kde se ukládají hesla v prostém textu a kdokoliv se k těmto datům dostane, tak si může zjistit hesla od řady uživatelů. Hesla by sice nikdy neměla být takto ukládána, ale je třeba si uvědomit, že internetové stránky může vytvořit kdokoliv a tedy i nejrůznější podvodníci. Podvodník může vytvořit věrohodný web o vaření a postupem času získá stovky zaregistrovaných uživatelů. Pokud takto cíleně špatně napíše aplikaci, tak si pak snadno může zjistit hesla jednotlivých uživatelů a zkoušet je na nejrůznějších místech, kde by se uživatel dal poškodit.

Samozřejmě to není jediný způsob, jak přijít o přihlašovací údaje. Tímto příkladem jsem chtěl ukázat, že by uživatelé neměli na internetu nikomu a ničemu věřit. Stačilo by, kdyby si lehkomyslný uživatel napsal přihlašovací údaje na papírek a ten se dostal do rukou komukoliv jinému, ale takových možností je nespočet. Ať tak či onak, více přihlašovacích jmen a hesel vyžaduje větší režii, ale zajišťuje to větší bezpečnost, takže opět je na každém z nás, co je pro něho důležitější.

### 3.4.4 Bezpečnostní otázky

Mnoho internetových služeb při registraci uživatele požaduje vyplnění bezpečnostní otázky, která slouží pro identifikaci při zapomenutí hesla. Otázky jsou většinou typu: *Jméno Vaší matky za svobodna?*, *Vaše číslo řidičského nebo občanského průkazu?*, *Jméno vašeho psa? atd.*, ale jsou tyto otázky opravdu bezpečné? Jméno vaší matky za svobodná může znát kdokoliv z nejbližších přátel, čísla občanského nebo řidičského průkazu můžete omylem poskytnout komukoliv a jméno vašeho psa zná každý kamarád nebo soused. Pokud je navíc možnost tuto informaci dohledat na internetu, tak takové otázky už nejsou vůbec bezpečné, tak jak na tuto problematiku vyzrát?

Jedním řešením je na tyto otázky odpovídat zcela nesmyslně např. na otázku ohledně jména vašeho psa odpovědět: *motorka*, pokud se ovšem váš pes nejmenuje motorka. Další možností by bylo, vytvořit si bezpečnostní otázku sám a na ní by znal odpověď pouze sám uživatel a nikdo jiný. S touto metodou jsem se ještě nesetkal, i když programově vytvořit tuto možnost není nijak náročné.

### 3.4.5 Ověřené Wi-Fi sítě

V dnešní době se neustálé rozmnožují *Wi-Fi* sítě, které už je možné najít takřka na každém rohu. Pokud tyto sítě uživatelé využívají, tak by měli být opatrní, zda je daná síť zabezpečená, obzvlášť při nakupování v *e-shopech* nebo při vstupování do internetového bankovnictví. Pokud daná *Wi-Fi* síť není zabezpečená a chráněná heslem, tak by jejich komunikaci mohl kdokoliv odposlechnout a zjistit si různé přihlašovací údaje nebo čísla platebních karet. Může se to zdát jako triviální záležitost, ale uživatelé by na to měli neustále pamatovat, protože stačí, když by jednou na to zapomněli a může nastat problém.

### 3.4.6 Zabezpečená komunikace

Dalším nutným základem při přihlašovacích a platebních procesech je zabezpečená komunikace. Dnes je tím docíleno převážně pomocí *SSL protokolu*, kterým je šifrována komunikace mezi uživatelem a serverem. Tento protokol pro šifrování používá digitální certifikáty, pomocí kterých se provádí autentizace a šifrování přenášených dat. Uživatel snadno pozná takto zabezpečené stránky podle *URL adresy*, která začíná prefixem *https*. Uživatel by měl být obezřetný při každém přihlášení a na jakýchkoliv stránkách zkontrolovat, zda dané stránky toto zabezpečení používají. Pro internetové bankovnictví a *e-shopy* by to mělo být nutností. Nepoužívají-li toto zabezpečení, tak není doporučeno se na takové stránky přihlašovat.

### 3.4.7 Vymazání cookies

Další možností pro zvýšení bezpečnosti je vymazání *cookies* po každém použití prohlížeče. Tuto možnost lze nastavit téměř v každém prohlížeči, aby po vypnutí prohlížeče *cookies* byly vymazány. Vymazáním sice uživatel ztrácí pohodlnost, protože vypne-li uživatel prohlížeč s tímto nastavením, tak při každém dalším spuštění musí opět např. zadávat přihlašovací údaje na již navštívených stránkách. Krom nastavení v prohlížeči existuje i řada programů, které jsou schopny internetové *cookies* vymazat. Uživatel se ale musí rozhodnout, jestli je důležitější pohodlnost a nebezpečí, nebo větší úsilí a větší bezpečnost.

## 3.5 Služba mojeID

Projekt *mojeID* [33] je bezplatná služba, která slouží k ověření uživatele na internetu s jeho skutečnou identitou. Ověření probíhá ve více fázích, kde je kontrolováno jméno, příjmení a také datum narození, čímž garantuje větší věrohodnost uživatelů, než je tomu u služeb, které žádnou podobnou službu nevyužívají. Této služby pak využívají jiné internetové aplikace, které ověření uživatelů vyžadují. Díky službě *mojeID* uživateli navíc stačí jedno přihlašovací jméno s heslem a pomocí *mojeID* je schopen se přihlašovat do všech služeb, které ho používají.

Služba *mojeID* tedy do jisté míry pravost uživatele zaručuje, ale v kapitole 3.4.3 píši o doporučení používat více uživatelských jmen a hesel, což vylučuje používání této služby. Ano, z pohledu větší bezpečnosti by tuto službu uživatel neměl používat, protože jediný úspěšný útok na uživatele bude pro něj znamenat velký problém. Útočník by najednou měl přístup ke všem službám, kde by byl pomocí *mojeID* zaregistrovaný a kde by nebyl, tam by mu mohl účet vytvořit a o zneužití už není potřeba psát.

# 4 Internetoví roboti/boti

*Internetový robot* (zkráceně jen *bot*) je počítačový program, který na internetu vykonává opakovanou činnost. Mezi tyto činnosti lze řadit vyhledávání informací, sbírání dat, odesílání informací nebo různé zpracovávání požadavků, kvůli nimž byl konkrétní robot stvořen. V následujících kapitolách tyto roboty klasifikuji, popíši, jak je na serveru detekovat a jak se před nimi chránit.

## 4.1 Klasifikace

Existují stovky známých robotů, kteří jsou veřejně známí [34]. Jedním z nejznámějších a zároveň nejlepších je *GoogleBot* [35], který používá vyhledávač *Google*. Pyšní se svou statistikou prohledaných internetových stránek, která přesahuje 95% všech stránek na internetu [36]. Číslo může být ve skutečnosti o něco menší, protože jak mohou lidé z *Googlu* tvrdit, že jim chybí prohledat necelých 5%? Když tento počet neprohledali, tak přece nemůžou vědět, kolik jim jich ve skutečnosti chybí.

Na druhou stranu existuje i řada robotů, o kterých nikdo neví a na internetu dolujou data podle potřeby majitele nebo cíleně zatěžují servery až k jejich pádů. Robotů existuje celá řada a liší se samozřejmě tím, co je jejich hlavním úkolem, at'už se jedná o roboty užitečné nebo ty, kteří mají škodit. V následujících kapitolách uvedu kategorie nejčastějších internetových robotů [37].

### 4.1.1 Vyhledávací robot

Tento druh robota, známý také pod anglickými slovy *crawler, spider* nebo *gatherer*. Robot tohoto druhu má na svém vstupu množinu internetových stránek, které prohledává a hledá na nich odkazy směřující na další stránky. Prohledávané stránky může zpracovat dle libosti, ale nejčastěji obsah stránky indexuje a vytváří z nich databázi, čehož využívají především internetové vyhledávače. Procházené stránky si ukládá, aby nedocházelo k jejím opakovaným prohledáváním. Po čase ovšem tyto stránky prohledává, protože v rámci časového období mohlo dojít ke změně obsahu nebo dokonce k odstranění stránky.

Samotný proces indexování je poměrně složitý a aby byl výsledek vyhledávání co nejlepší, tak se pro indexování používají různá kritéria, ze kterých dále uvedu ty nejpodstatnější.

- **Váha slova** - Prohledávaná stránka má větší šanci být ve výsledku vyhledávání, pokud obsahuje hledaný výraz s vyšší váhou. Vyšší váhu mají nejčastěji ty výrazy, které se na stránce opakují, jsou obsaženy v titulcích, nadpisech nebo jsou blíže k začátku stránky. Toto kritérium bývá zneužíváno umístěním často vyhledávaných výrazů na stránku, které ve skutečnosti nemají nic společného s jejím skutečným obsahem.
- **Sponzorované odkazy** - Různé osoby nebo firmy si můžou u vyhledávačů zvýhodnit své stránky zaplacením určitých poplatků. Tyto sponzorované stránky jsou pak zvýhodňovány při vyhledávání určitých výrazů. Vyhledávače si takto mohou přijít

na určitý zisk, ale zároveň tím riskují ztrátu svých uživatelů, kterým se nemusí líbit neadekvátní výsledky vyhledávání.

- **Atraktivita stránky** - Atraktivní stránkou je ta, na kterou vedou odkazy z více různých stránek, protože tato stránka pravděpodobně obsahuje zajímavý obsah. Toto kritérium lze zneužít vytvořením falešných stránek obsahující odkazy, které směřují na stránku, jež má být zvýhodněna.
- **Technická kvalita** - Kritérium, u kterého je důležité především dodržování standardů a správné sestavení stránek.
- **Seriózност serveru** - Servery, které obsahují velký počet kvalitních stránek, mohou být při vyhledávání díky tomuto faktu zvýhodňovány.

### 4.1.2 Udržovací robot

Prospěšnými roboty jsou taky udržovací roboti, kteří provádí sadu úkolů za účelem údržby serveru. Mezi úkoly těchto robotů patří mazání nefunkčních odkazů, kontrola dostupných souborů, různé indexování, oprava gramatiky aj. Tito roboti provádí údržbu většinou v době, kdy je server nejméně využívány, což bývá většinou v noci.

### 4.1.3 Chatterbot

Tento druh robota se v současnosti stává stálé víc populární. Jedná se o inteligentního robota, který se většinou snaží napodobit lidský faktor a uživatelům automaticky odpovídá na různé dotazy. Nejčastěji bývá tento robot nasazen na velkých *e-shopech* a pomáhá lidem s jejich dotazy. V dotazu robot detekuje klíčové slova a v závislosti na nich se snaží uživateli co nejlépe odpovědět podle připravených šablon. Někteří *chatterboti* nepoužívají jen různé šablony, ale učí se z obrovského množství minulých konverzací. Jedná se ovšem jen o robota, a ten nestačí na vyřešení všech specifických dotazů, a tak je lidský faktor v krajních případech stejně nezbytný.

### 4.1.4 IRC/ICQ Roboti

Tento druh robotů je velmi podobný *chatterbotům*, ale oproti nim jsou tito roboti považováni za škodlivé. Tito roboti odesílají převážně reklamy do různých diskuzí nebo odposlouchávají komunikace a v závislosti na ni, pak odesílají příspěvky. V případě *IRC robota* je komunikace odposlouchávána přes *IRC kanál*, z čehož vyplývá, že roboti můžou fungovat i mimo internetové stránky. *ICQ robot* je robot pro konkrétní aplikaci a jak už z názvu vyplývá, jedná se o chatovací aplikaci *ICQ*. Jeho funkce je obdobná jako u *chatterbotů* či *IRC robotů*. Náhodným uživatelům odesílá různé reklamy a odkazy na zavirované stránky. Ti chytřejší *ICQ roboti* dokážou simulovat i lidskou komunikaci jako *chatterbot*.

### 4.1.5 Spambot

Další kategorií jsou *spamboti*, tento název vznikl ze slova *spam*, což je ve světě většině případů nevyžádaná elektronická pošta, která má reklamní podtext. *Spam* se ovšem vyskytuje i v různých internetových diskuzích a na mnoha internetových serverech, kde je možnost vložení nějakého příspěvku. Tuto možnost *spambot* detekuje a automaticky se zde pokusí vložit příspěvek. *Spambotu* je dnes celá řada, které lze klasifikovat do následujících kategorií:

- **E-mailový spambot** - Hlavní dominantou tohoto robota je procházení internetu a hledání *emailových* adres, které jsou snadno identifikovatelné pomocí znaku zavináče *(@)*, který je obsažen v každé *emailové adrese*. Na získané *emailové adresy* jsou pak většinou zasílány reklamní *emaily*, o které majitel *emailu* vůbec nestojí. Kvalitnější internetové servery své uživatele chrání před takovýmto odchytem *emailových adres* různými způsoby. Jedním z nich může být použití obrázku, ve kterém je *email* vepsaný. Další možností bývá nahrazení znaku zavináče různými textovými variacemi, tak, že *email bbb@ccc.com* může mít např. následující podoby.

  - bbb (zavináč) ccc.com
  - bbb (at) ccc.com
  - bbb (ALT + 64) ccc.com
  - bbb (ALT + V) ccc.com

  Uvedené textové variace do jisté míry chrání uživatele před odchytem *emailové adresy*, ale z pohledu robota v tom nevidím zas takový problém. Pokud robot tyto variace zná, tak je může vyhledávat stejně tak jako zavináč.
  Dále existují také spamboti, kteří provádějí slovníkový útok na poštovní servery používáním nejčastějších jmen, přezdívek a jejich různých kombinací. Při útoku sleduje odpovědi *SMTP serveru*, zda byl *email* doručen. V případě úspěchu je vytvořený *email* uložen a v budoucnu na něho může být zaslán reklamní *email (spam)*.

- **Fórum spambot** - Tito spamboti prohledávají internet a především internetové fóra nebo stránky, kde je možnost vložení komentáře. Na těchto stránkách se pak snaží vložit vlastní komentář, který většinou něco propaguje. Ochrana proti těmto *spambotům* při vkládání příspěvků spočívá v položení kontrolních otázek nebo použití ověřovací metody nazývané *captcha*, kde se jedná o přepis textu z různě zdeformovaného obrázku. Na tyto techniky už robot nedokáže logicky odpovědět a případný příspěvek není vložen. V případě *captchy*, by bylo možné zdeformovaný text detekovat díky *technolgii OCR*, ale vzhledem k různorodosti obrázku na různých serverech je to ve větší míře neaplikovatelné.

### 4.1.6 Botnet

*Botnet* je velká síť propojených robotů, které infikovaly mnoho počítačů. Tito roboti jsou součástí počítačových virů a pokud tento robot infikoval nějaký počítač, jeho majitel o

tom většinou nemá ani ponětí. Tito roboti jsou pak na dálku ovládáni jeho majitelem, nejčastěji na rozesílaní spamů nebo *DDoS útoky*. Podobně jako *DDoS útok* funguje taky *DoS útok*. Rozdíl je pouze v tom, že v případě *DoS útoku* pochází útok pouze od jediného stroje (jedince). U *DDoS útoků* se jedná o dva a více (mnohdy i sta-tisíce) strojů.

## 4.2 Detekce

Roboti můžou serverům působit poměrně velké problémy, pokud na něho neustále posílají dotazy nebo zahlcují diskuzní fóra nesouvisejícími příspěvky. Server je tímto zbytečně vytížený a omezuje jeho normální běh, ke kterému je určen. Při vkládání různých příspěvků robotem, je navíc poškozen obsah stránky. Provozovatelům serverů se tyto praktiky pochopitelně nelíbí a tak existují různé metody, jak takové roboty detekovat a znemožnit jim tyto činnosti. Konkrétních řešení existuje spousta, ale principů na základě kterých je to možné, je pouze několik, které níže uvedu.

### 4.2.1 Mnoho dotazů (requestů)

Pro roboty je typické příliš časté dotazování na servery. Pokud mají k dispozici dostatek odkazů na daný server a nemají nastavenou žádnou prodlevu, tak takové dotazování probíhá velice rychle. Pokud stránky nejsou přehnaně veliké *(>1 MB)* a jejich stažení netrvá dlouho, tak se jedná o maximálně stovky milisekund. Samozřejmě rychlost komunikace mezi serverem a klientem vždy závisí na rychlosti připojení k internetu. Detekuje-li server, že z jedné *IP adresy* byl např. během 1 sekundy 5x dotázán na odpověď. Pak lze snadno prohlásit, že se jedná o robota, protože není možné, aby se člověk stránkami takto rychle proklikal. Do této kategorie spadají především internetoví roboti, *DoS a DDoS útoky*, které se vyznačují taktéž obrovským množstvím dotazů na server.

### 4.2.2 Periodicita

Sofistikovanější roboti obejdou předešlý bod 4.2.1 nastavením určité periody, ve které se dotazy opakují. Pokud server kontroluje např. v jakých periodách je dotazován z jedné *IP adresy*, tak může být pro provozovatele serveru podezřelé, že z jedné *IP adresy* byl dotázán např. 10x a každý dotaz přišel v intervalu 23 sekund $\pm$ 100 milisekund (22.9s, 23.1s). Uvedená rezerva 100 milisekund má svůj význam, protože jednotlivé dotazy na server prakticky nikdy nebudou trvat stejnou dobu, díky *latenci sítě* a právě tato rezerva by měla pomoci detekovat opakující se periodu. Záleží už pak jen na konkrétním provozovateli serveru, jak velkou prodlevu zvolí pro tuto detekci zvolí.

### 4.2.3 User-Agent

Poměrně jednoduchou metodou je zjištění robota na základě *User-Agenta*. Ten je součástí *Http requestu* a obsahuje určité informace o klientských aplikacích a jejich různých verzí. Princip je jednoduchý, pokud *User-Agent* obsahuje nějaké jméno známého robota z databázové tabulky robotů, tak lze konstatovat, že se jedná o robota.

### 4.2.4 Rychlé vyplňování formulářů

Obsahuje-li stránka různé formuláře pro vkládání příspěvků, tak zde připadá v úvahu odhalit robota typu *spambot* 4.1.5, který se pokouší o vložení různých příspěvků. *Spambot* automaticky vyplňuje jednotlivé položky, ale provádí je mnohem rychleji, než-li by je vyplňoval člověk. Pomocí tohoto by tedy bylo možné robota detekovat. Pro tuto detekci by šlo použít již uvedené možností, jakými jsou *rychlé dotazy* a *periodicita*.

## 4.3 Prevence

Některým provozovatelům nemusí vadit, že jejich server skenuje nějaký robot, protože to oni právě chtějí, aby je roboti zaindexovali a jejich internetové stránky se tak objevovaly ve výsledcích vyhledávání. Takový skenující robot může vést až k pádu severu nebo velkému vytížení, což ho rapidně zpomalí. Proto vznikly různé důležité techniky, jak ochránit svůj server, o kterých se dále zmíním.

### 4.3.1 Síťové zařízení

Před *DoS a DDoS útoky* [41] je nastavení síťových prvků *(Firewall, Router, Intrusion Detection System, Switch, aj.)* jednou z nejdůležitějších součástí zabezpečení. Základem je instalace nejnovějších aktualizací, které odstraňují nalezené problémy. Veškerá komunikace by se měla logovat a v případě anómálií se dopátrat, jak, odkud a proč nastaly konkrétní problémy. Ty pak můžou být pomocí logů jasně detekovány a na základě nich se můžou udělat nová zaopatření. Nastavení na síťových zařízeních je skutečně mnoho, ale to není součástí této práce.

Za zmínku ale stojí *SYN cookies*, které jsou hlavním obranným prvkem při obraně před *DoS útoky*. Interně modifikují chování *TCP protokolu* tak, že je server klientovi přístupný až po ověření platnosti adresy.

### 4.3.2 Servery/Aplikační servery

Další možnosti jak se bránit útokům je ochrana přímo na serverech, kde běží internetové stránky. Existují zde různé způsoby jak zabránit spojení s koncovým klientem. Lze zde taktéž nastavovat různé *firewally*, ověřovací certifikáty nebo definovat *access listy*, což jsou seznamy *IP adres*, kde jsou definovány povolené popř. zakázané *IP adresy*.

V poslední řadě lze nastavit aplikační servery, které poskytují taktéž různá omezení klientských požadavků. V obou případech je více možností a vždy záleží na konkrétních technologiích, které poskytují své specifické vlastnosti. Ani v těchto případech by se nemělo zapomínat na aktualizace softwaru, které odstraňují bezpečnostní chyby.

### 4.3.3 Robots.txt

Soubor *robots.txt* [42] umožňuje autorům internetových stránek specifikovat, ke kterým stránkám by robot neměl přistupovat. Musí být umístěn v kořenovém adresáři webu a slouží pouze k nastavení pravidel pro stahování stránek.

Každý slušný robot, který chce prohledávat konkrétní server, tak by si jako první věc měl projít soubor *robots.txt* a zjistit, které stránky si provozovatel nepřeje skenovat. Realita je ale taková, že tento soubor většinou brán v potaz není a robot prohledává celý server.

# 5 Implementace vlastního robota

Součástí této kapitoly je popis mnou vytvořené aplikace, kterou jsem se rozhodl pojmenovat *robot7*. Aby nedocházelo k záměně obecného internetového robota a mnou vytvořené aplikace, rozhodl jsem se zakončit název číslem 7 *(robot7)*. Bude-li tento název dále uveden, tak je tím myšlena právě mnou vyvinutá aplikace.

Úkolem bylo vyvinout program, který bude monitorovat *digitální stopu* a následně graficky znázorňovat výsledky. Vytvořil jsem tedy internetového robota, který spadá do kategorie vyhledávacích robotů 4.1.1. Výsledkem mé práce je desktopová aplikace, která potřebuje jako vstup internetovou stránku spolu s dalšími možnostmi, které uživatel může na začátku specifikovat. Po spuštění začne *robot7* na zadané stránce hledat odkazy na další internetové stránky a na těchto stránkách další odkazy. Tímto způsobem *robot7* pokračuje, dokud se nezastaví na základě nějaké vstupní podmínky nebo ho nezastaví sám uživatel, protože toto vyhledávání obvykle nezná konce. Stačí jediný odkaz na známější server typu *www.yahoo.com* a z toho už vedou odkazy na podstatnou část internetu. Po dokončení celé akce je výsledkem strom odkazů, jež byly nalezeny, dále pak statistika vyhledávání, log obsahující vypsané chyby při vyhledávání, možnost vykreslení grafu spolu s dalšími možnostmi zobrazení. Výsledný graf je možno uložit do souboru *xml* a poté ho kdykoliv znovu v aplikaci zobrazit za pomocí tohoto souboru.

## 5.1 Použité technologie

Pro implementaci jsem použil technologie, které jsou licencované jako *open-source* a jsou zcela zdarma. Nebylo potřeba použití placených technologií a různých placených doplňků, i když v případě vykreslování grafů by bylo použití placených knihoven možné pro jiné grafické znázornění. V mém případě to nebylo potřeba a mnou použitá knihovna pro grafy svými možnostmi plně dostačuje. Výslednou aplikaci představuje spustitelný *jar* soubor.

### 5.1.1 Java

Pro *robota7* jsem použil objektově orientovaný jazyk *Java* [38], jelikož je rozšířen po celém světě a nabízí spoustu možností řešení různých problémů. Pro *robota7* je důležitá především síťová komunikace a ta je v *Javě* na vysoké úrovni. Dále bylo potřeba mít framework pro uživatelské rozhraní *robota7* a pro ten jsem použil knihovnu *Swing*, která je součástí standardní verze *Javy*. Veškerý vývoj tedy probíhal v *Javě* a použil jsem pro něj vývojové prostředí *Eclipse* [39].

### 5.1.2 Apache Maven

*Apache Maven* [40] zkráceně jen *Maven*, je buildovací nástroj, který slouží k automatizaci a sestavování výsledných buildů aplikace. Je rozšířen pro více programovací jazyků, ale převážně je podporován pro *Javu*. Pomocí *Mavenu* lze snadno specifikovat závislosti na knihovnách třetích stran a přidávat různé doplňky (pluginy). Pro všechny důležité

informace o sestavení projektu slouží soubor *pom.xml*, který je součástí každého *Maven* projektu. V *robotovi7* jsem použil následujících přídavků.

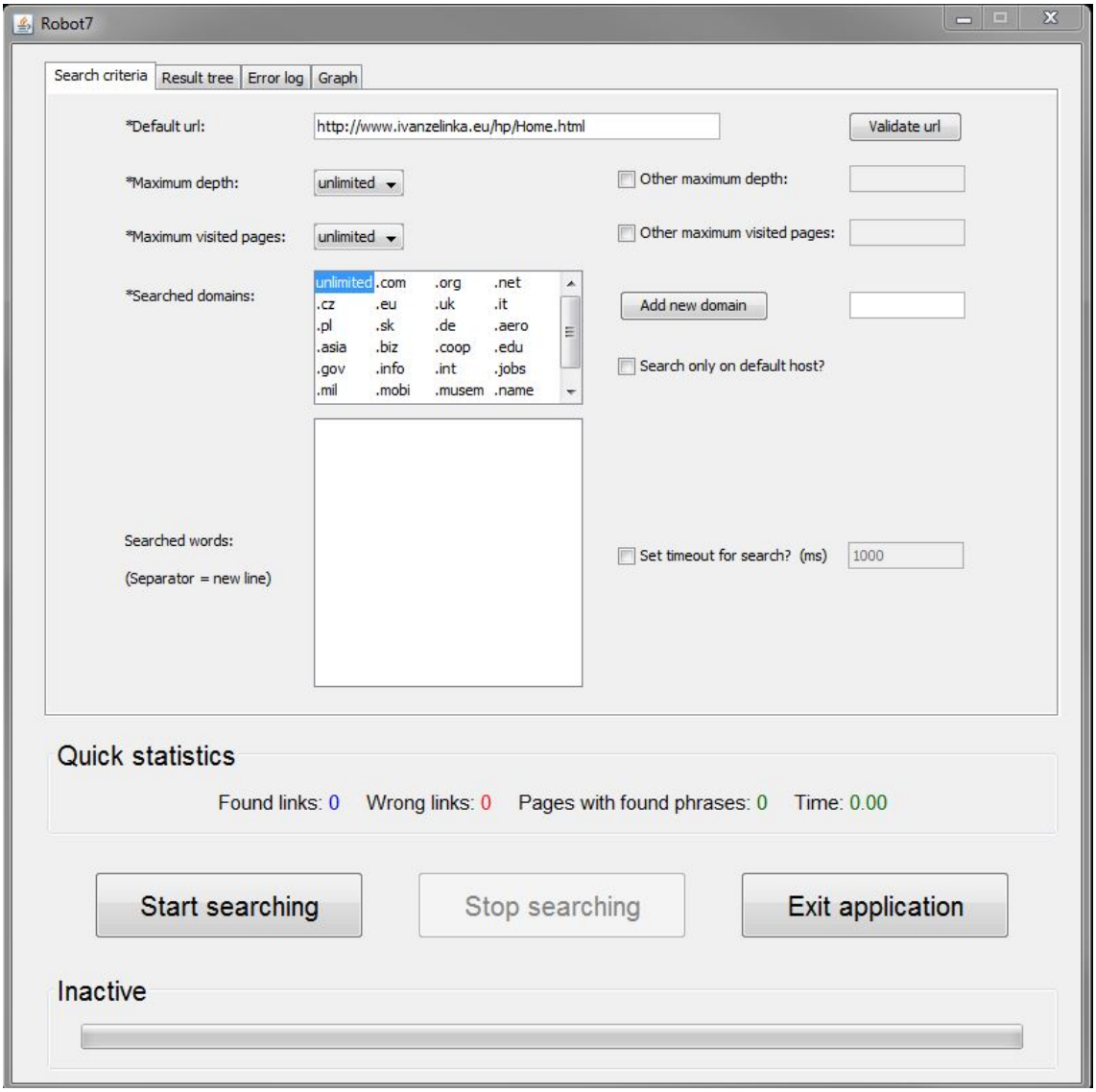
- **Doplňky (plugins)**
  - **Apache Maven JAR Plugin** - Tento plugin slouží ke specifikování hlavní třídy, pomocí které se celá aplikace spouští. Hlavní třída v *Javě* musí obsahovat metodu *main*, která je volána jako první. V mém případě se jedná o třídu *Main.java*.
  - **Apache Maven Shade Plugin** - Tento plugin zajišťuje přidání knihoven třetích stran do výsledného *jar* souboru. Není-li tento plugin v *Mavenu* specifikovaný, tak build sice v pořádku projde a aplikace půjde spustit, ale dojde-li k volání nějaké třídy z knihovny třetí strany, tak nebude nalezena a aplikace spadne. Dále tento plugin používám pro specifikování názvů jednotlivých buildů a v poslední řadě pomocí něho minimalizuji výsledný *jar* soubor, což je velmi užitečná vlastnost. Je-li minimalizace zapnuta, tak do výsledného souboru (v mém případě *jar*) jsou přidány pouze ty třídy, které jsou v aplikaci použity a nikoliv celé knihovny, jak je tomu v normálním případě. Tímto lze často ušetřit mnoho místa, protože z knihoven se někdy používá pouze pár tříd.
- **Závislosti na knihovnách třetích stran (dependencies)**
  - **Apache Commons Validator** - Tato knihovna obsahuje řadu různých validátorů. Použil jsem pouze validátor pro validaci *URL adres*.
  - **Jsoup** - Knihovna *jsoup* slouží k parsování *HTML stránek*, což je v *robotovi7* potřebné pro hledání a nalezení odkazů. Dále jsem tuto knihovnu využil při indexování slov na *HTML stránce*, protože pro vyhledávání bylo potřeba získat pouze slova bez *HTML tagů*.
  - **Jung2** - Velice povedená kniha na tvorbu grafů, která umožňuje vytvářet grafy různých vzhledů. Jednotlivé grafy lze velmi dobře konfigurovat, ale je potřeba se dobře seznámit s programátorským *API*.

## 5.2 Uživatelské rozhraní

Jak už jsem dříve zmínil, tak *robot7* je z hlediska grafického rozhraní napsán ve *Swingu*. Hlavní okno je rozdělené na dvě části. První a zároveň vrchní část tvoří záložky (taby), mezi kterými uživatel může přepínat a sledovat různé akce. Spodní část tvoří hlavní ovládací panel, kde se nachází ovládací prvky *robota7*. Jsou tam tři tlačítka, pomocí kterých se aplikace ovládá. Dále tam jsou statistické údaje ohledně vyhledávání a na konci je *status bar*, který signalizuje *aktivní/neaktivní* stav *robota7*. Spodní část při přepínání záložek zůstává vždy viditelná, aby bylo možné *robota7* odkudkoliv ovládat. Všechny tyto komponenty jsou dále podrobně popsány.

### 5.2.1 Záložky (taby)

Po spuštění *robot7* vypadá tak, jak je zobrazeno na obrázku 5. Ve vrchní části jsou vidět čtyři záložky a po spuštění je vždy aktivní záložka s nastavením vyhledávacích kritérií (Search criteria).




Obrázek 5: Spuštění aplikace

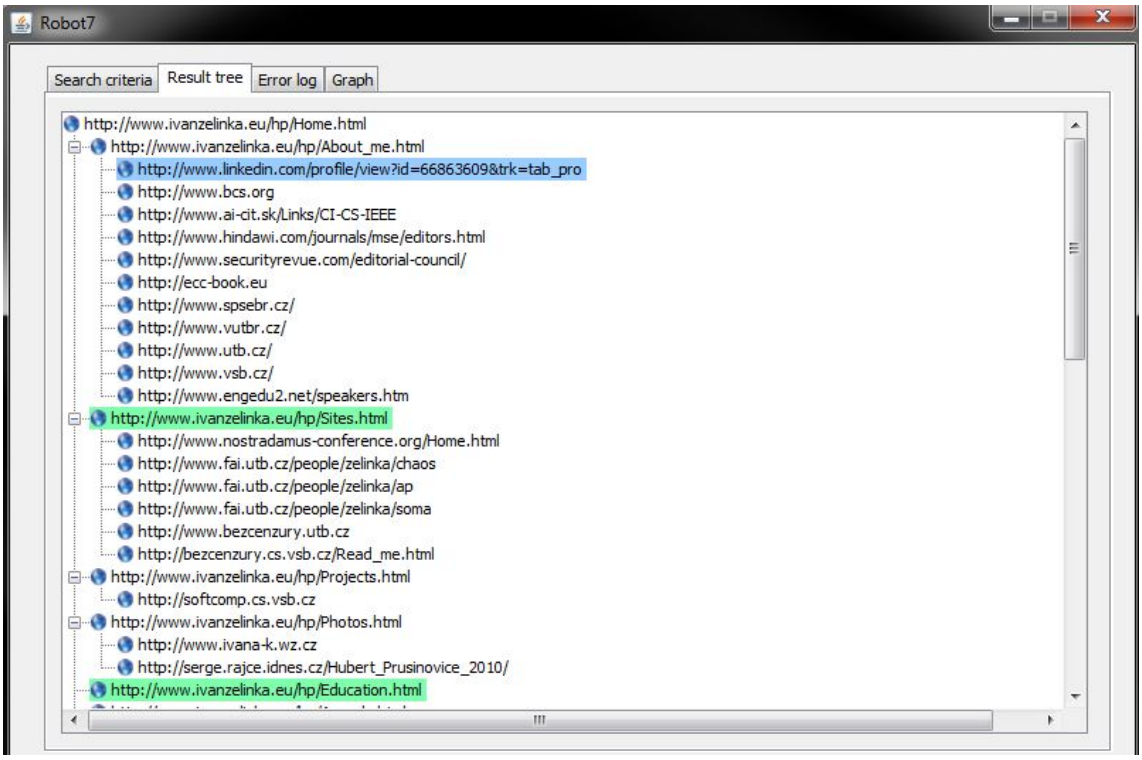
- **Vyhledávací kritéria (Search criteria)** - Na této záložce uživatel specifikuje všechny podstatné údaje pro vyhledávání *(crawlování)*, které jsou vidět na obrázku 5. Tyto údaje mají své omezení, které dále popíši.

- **Defualt url** - Zde uživatel vkládá *URL adresu*, na které začíná *robot7* vyhledávat. Adresu je možno před spuštěním ověřit tlačítkem *Validate url*, zda je validní. Pokud adresa není ověřena a není ani validní, tak vyhledávání stejně nezačne a uživatel je vyzván o opravu *URL adresy*.
- **Maximum depth** - Specifikuje, do jaké hloubky se má provádět vyhledávání. Pokud tedy bude chtít uživatel prohledávat stránky pouze na stránce, kterou uvedl výše, tak musí vybrat nebo zadat číslo 1. Chce-li prohledat ještě stránky, které se skrývají na stránkách nalezených ve hloubce 1, tak musí maximální hloubku nastavit na 2. Výchozím stavem je neomezená hloubka (*unlimited*).
  Při zadávání maximální hloubky má uživatel dvě možnosti. První možností je vybrání z předem vyplněného listu, který načítá hodnoty ze souboru *maximumDepth.txt*, který je ve složce *properties* a ta je uvnitř spouštěcího *jar souboru*. Výchozí hodnoty v souboru jsou: *unlimited, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 50, 100, 150, 200, 300, 400, 500, 1000* a uživatel je může přizpůsobit dle potřeb editováním tohoto souboru. Podmínkou je, aby každá hodnota byla uvedená na jednom řádku. Počet hodnot v souboru je neomezený.
  Druhý způsob specifikování je zaškrtnutí políčka *Other maximum depth* a uživateli je pak povolena možnost specifikovat vlastní hodnotu do zviditelněného políčka na stejném řádku. Povoleno je vložení pouze čísel $> 0$.
- **Maximum visited pages** - Tímto atributem uživatel specifikuje maximální počet stránek, kolik má *robot7* prohledat. Možnosti zadávání hodnot jsou zcela stejné, jak je tomu v předešlém případě. Rozdíl je akorát v původních hodnotách, které jsou: *unlimited, 10, 20, 30, 50, 100, 150, 200, 300, 400, 500, 1000, 2000, 3000, 4000, 5000, 10000, 50000, 100000* a soubor, z kterého se načítají, má název *maximumPages.txt*. Soubor se nachází opět ve složce *(properties)*, která je součástí *jar souboru* a tyto hodnoty může uživatel opět editovat dle potřeb.
  Pokud chce uživatel jednorázově zadat maximální počet stránek, tak zaškrtne *Other maximum pages* a tím je mu povolena možnost vložení jiné hodnoty.
- **Searched domains** - Další možnost, jak specifikovat vyhledávání, je vybrání domén, které mají obsahovat nalezené odkazy. Všechny domény jsou v panelu s doménami a uživatel jich může vybrat více najednou. Po spuštění aplikace je automaticky vybrána doména *unlimited*, což představuje vyhledávání bez omezení domén. Všechny hodnoty jsou načítány ze souboru *domains.txt*, který se nachází na stejném místě jako oba předchozí soubory, které se používají pro načítání hodnot. Původní načtené hodnoty jsou: *unlimited, .com, .org, .net, .cz, .eu, .uk, .it, pl, .sk, .de, .aero, .asia, .biz, .coop, .edu, .gov, .info, .int, .jobs, .mil, .mobi, .musem, .name, .pro, .tel, .travel, .xxx*. Domény lze opět přizpůsobit dle potřeb editováním tohoto souboru, ale uživatel musí být opatrný a pamatovat na to, že doména musí začínat tečkou a musí obsahovat minimálně dva znaky.
  Pokud se doména nenachází v panelu pro jejich výběr, tak je tu ještě druhá možnost, a to vložení domény do tohoto panelu přímo v aplikaci. To se provede vyplněním políčka, které je na stejném řádku jako panel s doménami. Poté se stiskne tlačítko *Add new domain*. Při zadávání je nutno dodržet dvě podmínky,

které mají zajistit větší pravděpodobnost správnosti nové domény. Pro vložení musí doména začínat tečkou a obsahovat minimálně dva další znaky. Pokud nejsou obě podmínky splněny, tak je uživatel informován o problému a doména není přidána. Takto přidána doména zůstane v aplikaci jen do jejího ukončení. Po opětovném spuštění už tato doména v panelu domén nebude.
Na stejném řádku jako je panel s doménami, se nachází ještě možnost zaškrtnutí políčka *Search only on default host?* Pokud ji uživatel zaškrtne, tak ve výsledku budou hledány odkazy pouze na stejném *hostu*, jako je výchozí *URL adresa*. Tato možnost je velmi užitečná, pokud chce uživatel prohledat konkrétní server. Tímto nejsou brány v potaz odkazy, které směřují na jiný server. Pokud uživatel použije jako výchozí *URL adresu* homepage daného serveru, tak je tímto způsobem získána mapa stránek prohledávaného serveru.

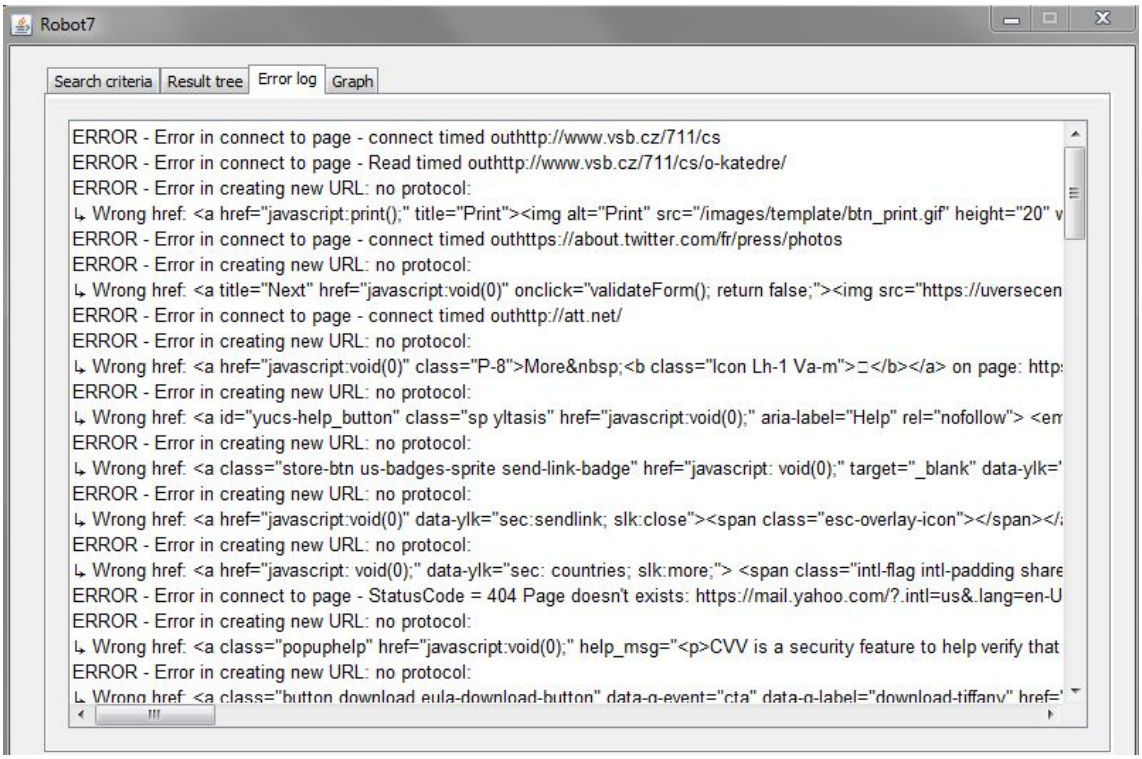
– **Searched words** - Zde uživatel zadává slova, které chce vyhledávat. Počet zadaných slov není omezen a pokud chce uživatel vyhledávat více slov, tak podmínkou je, aby každé slovo bylo na jednom řádku. Vyhledávání je dále podrobněji popsáno v kapitole 5.3
– **Set timeout** - Některé servery kontrolují počet dotazů jednotlivých uživatelů a při překročení určitých podmínek můžou zablokovat klientskou *IP adresu*. To lze v *robotovi7* částečně obejít zaškrtnutím pole *Set timeout for search? (ms)*. Tím je uživateli povolena možnost vložení času v milisekundách, který představuje prodlevu mezi jednotlivými dotazy. Výchozí hodnotou je *1000 (1 sekunda)*. Vkládat je povoleno pouze celá čísla.

- **Strom výsledků (Result tree)** - Na obrázku 6 je zobrazena druhá záložka, na kterou se aplikace automaticky přepne po spuštění vyhledávání. Úplně prvním a zároveň kořenovým uzlem stromu je odkaz na stránku, která byla zadána na prvním tabu (*Search criteria*) jako výchozí stránka pro vyhledávání. Zanoření jednotlivých odkazů (existujících stránek) znázorňuje, kde byly stránky nalezeny.
  Na obrázku 6 je vidět dva odkazy s různě barevným pozadím. Pokud je pozadí zelené, tak to signalizuje stránku, na níž se nachází hledané slovo alespoň jednou. Po kliknutí levým tlačítkem myši na odkaz se jeho pozadí obarví modře. Dvojklikem na odkaz se vybraná stránka otevře v uživatelově výchozím prohlížeči.



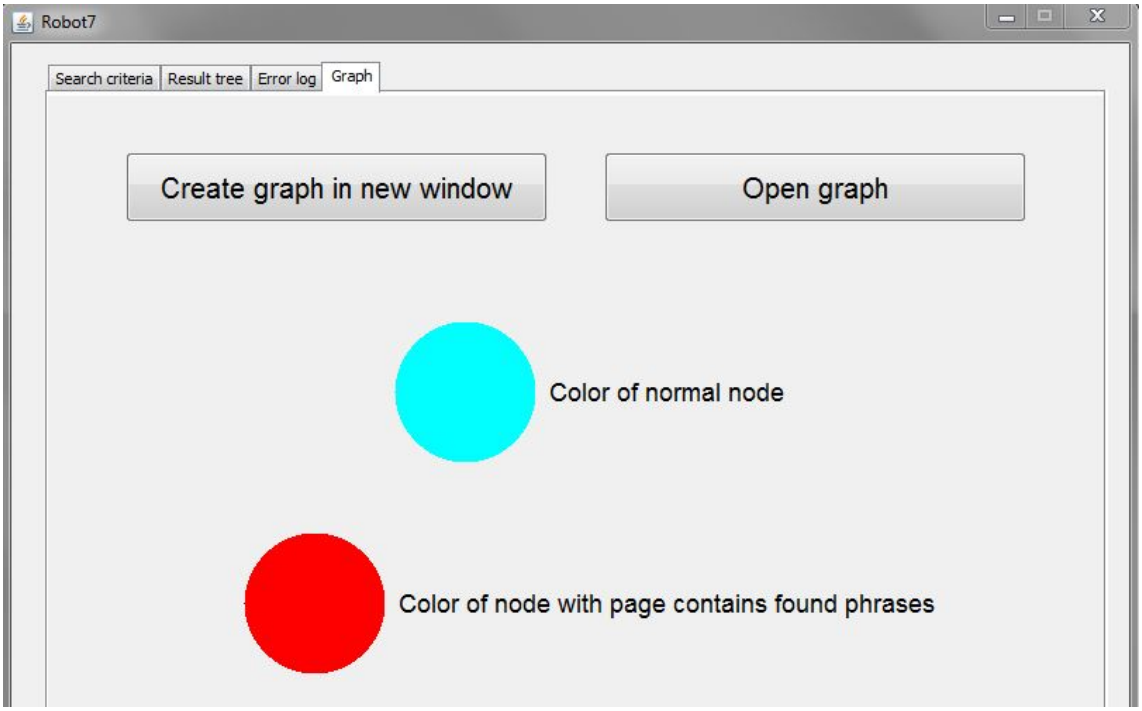

Obrázek 6: Výsledný strom obsahující odkazy

- **Chybový log (Error log)** - Tento tab zobrazen na obrázku 7 loguje chybové zprávy, pokud není nalezený odkaz v pořádku. Důvodů může být několik, ale zpravidla se nejčastěji jedná o následující problémy.
  - Nalezený odkaz je sice v pořádku, ale nejedná se o odkaz na internetovou stránku, ale na stažení souboru. To je pro můj *robot7* nežádoucí a nalezený odkaz tedy není přidán do výsledného stromu.
  - Vyprší čas pro stažení internetové stránky. Zde je nalezený odkaz v pořádku, ale z nějakého síťového problému se stránku nepodaří v daný čas stáhnout. Proto se někdy stane, že i po vyzkoušení odkazu z chybového logu je stránka v prohlížeči zcela v pořádku otevřena.
  - Klasickým problémem je již neexistující stránka. Odkaz na stránku je v pořádku, ale stránka už neexistuje. Server odpovídá *chybou 404*.
  - Posledním problémem, který se mi při vyhledávání objevil, je odkaz na stránce, který neodkazuje na další stránku, ale pod tímto odkazem se volá nějaká *javascriptová funkce*. Funkce už může dělat cokoliv, např. mění písmo na stránce, ale pro *robota7* je toto nepodstatné, protože to není odkaz na stránku.




Obrázek 7: Chybový log

- **Graf (Graph)** - Na této záložce 8 jsou dvě tlačítka. Tlačítko *Create graph in new window*, kterým se vytvoří v novém okně graf 5.2.2, ten jsem se rozhodl vytvořit v novém okně, protože vytvářeny graf je poměrně velký a přímo na záložce by byl malý. A druhým tlačítkem je *Open graph*, které po kliknutí zobrazí dialog pro vybrání a následné zobrazení již uloženého grafu v souboru *xml*.
  Dále se na této záložce nachází dva barevně vyplněné kruhy s popisem, jaký význam v grafu mají jednotlivé barvy. Zobrazené kruhy představují v grafu jednotlivé uzly a každý uzel je jedna stránka (odkaz). Tyrkysová barva kruhu znamená, že internetová stránka neobsahuje ani jedno hledané slovo. Červený uzel signalizuje, že se na něm (stránce) vyskytuje aspoň jedno hledané slovo.
  Ke každému červenému uzlu vede od kořenového uzlu cesta červených hran. Tímto může uživatel v grafu vyhledat cestu ke stránce, na níž se nachází hledané slovo, protože má-li nějaký uzel mnoho tyrkysových potomků a červený uzel je např. ještě o pět úrovní níž, tak nejde vůbec vidět (v případě rozvržení *balón*) a je překrytý uzly s tyrkysovou barvou. Proto jsou hrany vedoucí k výsledku takto obarveny. Příklady obarvení hran lze vidět na obrázcích 9, 10 a 11. Každý uzel obsahuje i číselný popisek, který říká, kolik hledaných slov se na daném uzlu a všech jeho potomcích nachází. Po dvojkliku myši na kterýkoli uzel, se uživateli otevře internetová stránka ve výchozím prohlížeči, jež daný uzel reprezentuje. Stránku, kterou uzel reprezentuje lze vidět po najetí kurzoru myši na daný uzel. 10

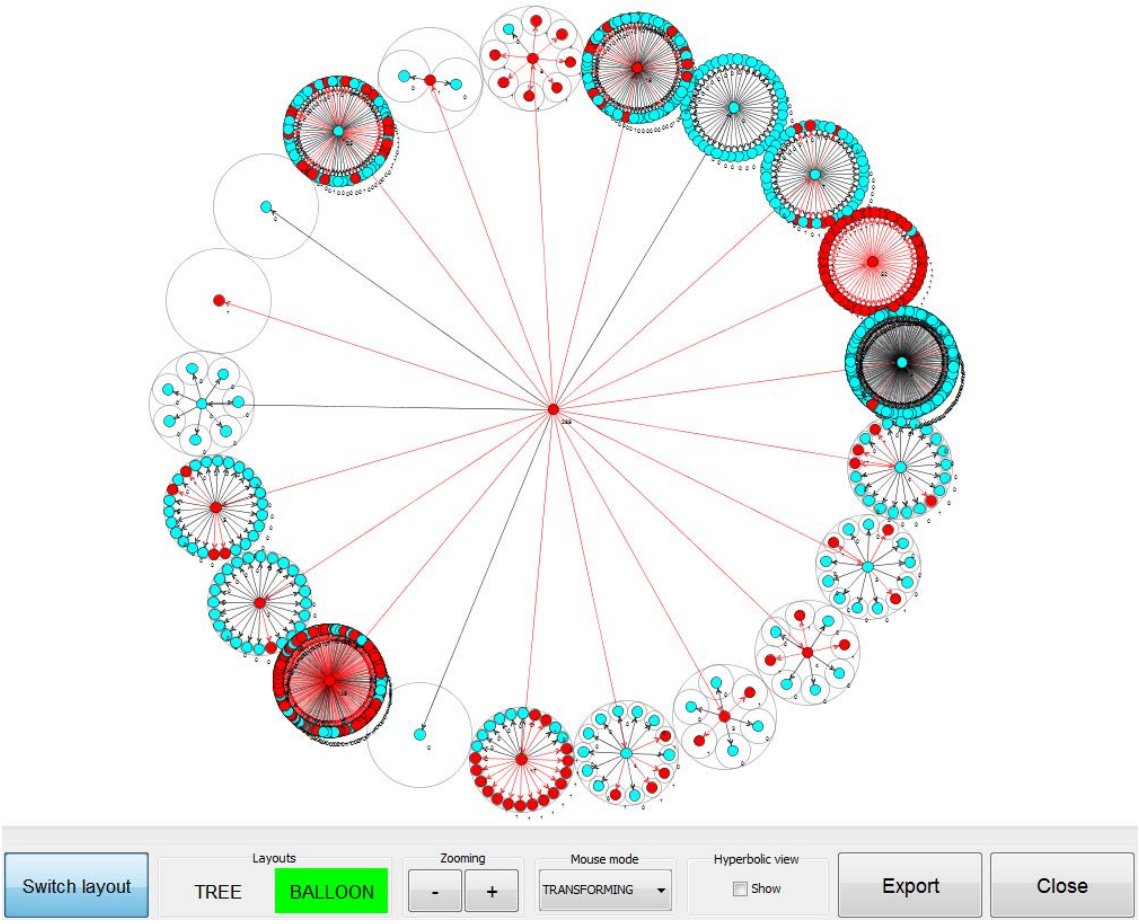



Obrázek 8: Graf tab

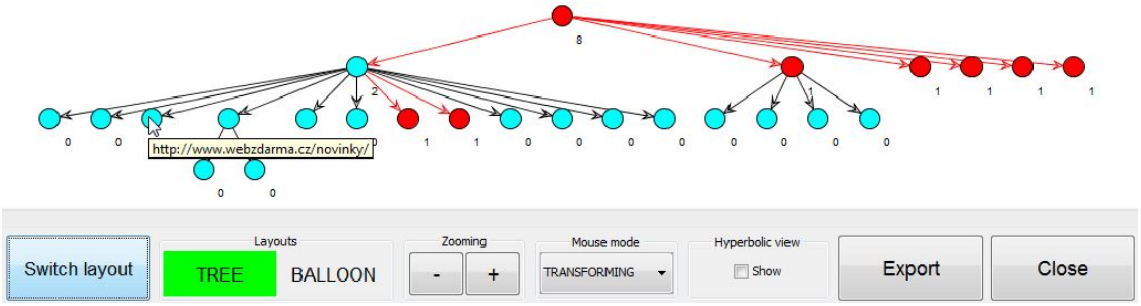
### 5.2.2 Okno grafu (Graph window)

Okno s vytvořeným grafem je znázorněno na obrázku 9 a 10. Okno obsahuje několik ovládacích prvků, které budou dále popsány.

Prvním ovládacím prvkem je tlačítko *Switch layout*, které mění rozvržení grafu. Původně jsem *robota7* napsal s rozvržením *strom (tree)*, které je vidět na obrázku 10. Bohužel toto rozvržení nemá smysl pro obrovský počet uzlů (nalezených stránek), což je pro *robota7* specifické. Proto jsem do aplikace přidal rozvržení *balón (balloon)*, na kterém jsou uzly uspořádány do kruhu a jednoduchým přibližováním v grafu se uživatel může zanořovat, tak jak by jednotlivé stránky proklikával. Výchozím rozvržením je *balón*.




Obrázek 9: Okno grafu s rozvržením *balloon*

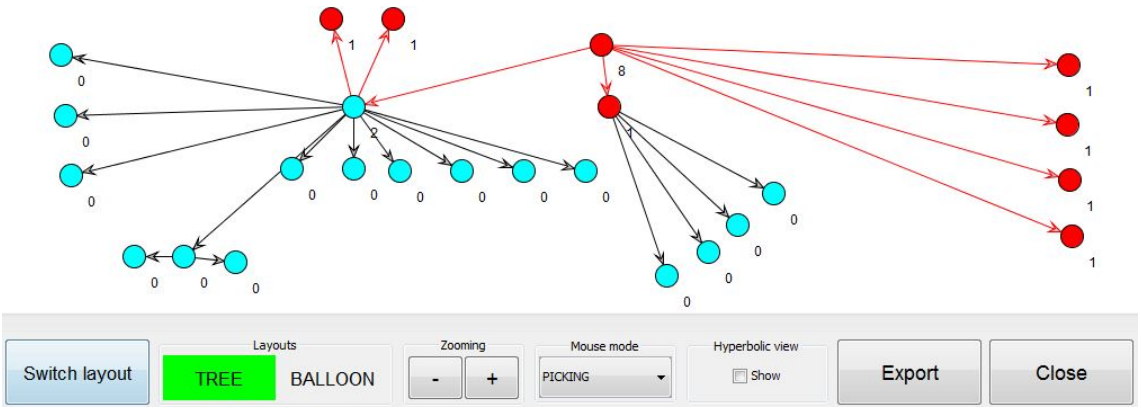



Obrázek 10: Okno grafu s rozvržením *tree*

Součástí ovládacích prvků je dále zvýraznění aktivního rozvržení zeleným pozadím v rámečku *rozvržení (Layouts)*.

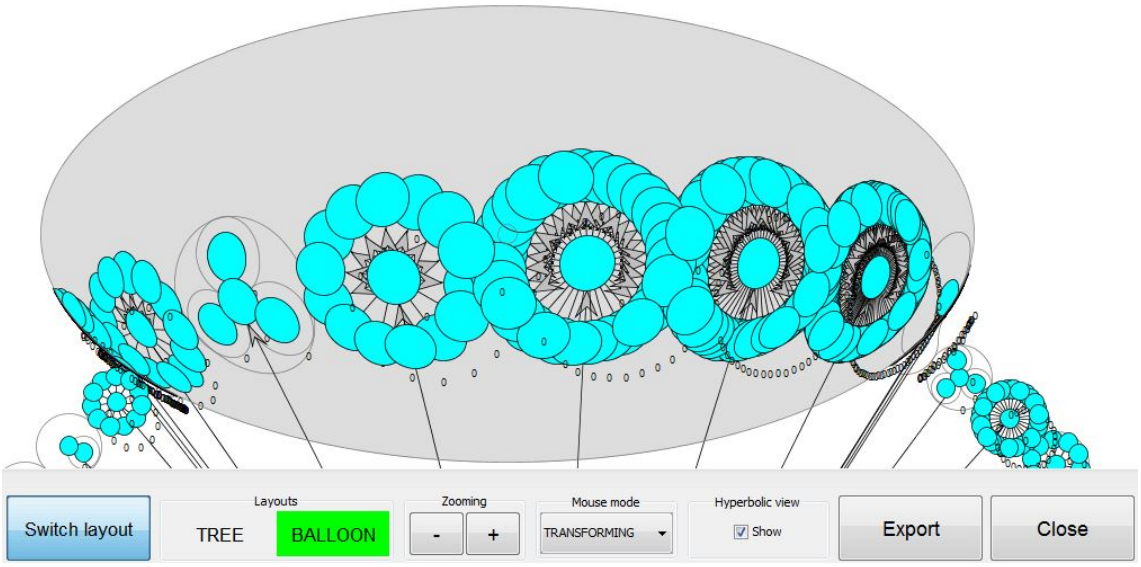
Další ovládací prvky jsou v rámečku přibližování *(Zooming)*, kde jsou tlačítka + a -, které přibližují nebo oddalují pohled na graf.

Následujícím rámečkem je *mód při pohybu myši (Mouse mode)*, kde má uživatel dvě možnosti. První je *transformační (Transforming)*, čímž uživatel při stisknutí a držení levého tlačítka myši posouvá celý graf. Druhým módem je *vybírání (Picking)*, který uživateli umožňuje jednotlivé uzly grafu přemístit dle potřeby. Příklad takového přemístění je vidět mezi obrázky 10 a 11.




Obrázek 11: Okno grafu s rozvržením *tree* s využitím *PICKING*

Další možností na tomto okně je zaškrtnutí položky *zobrazit (Show) hyperbolické zobrazení (Hyperbolic view)* v posledním rámečku. Toto zobrazení funguje podobně jako lupa částečně přibližuje graf tak, jak je znázorněno na obrázku 12.




Obrázek 12: Zobrazování grafu s použitím *Hyperbolic view*

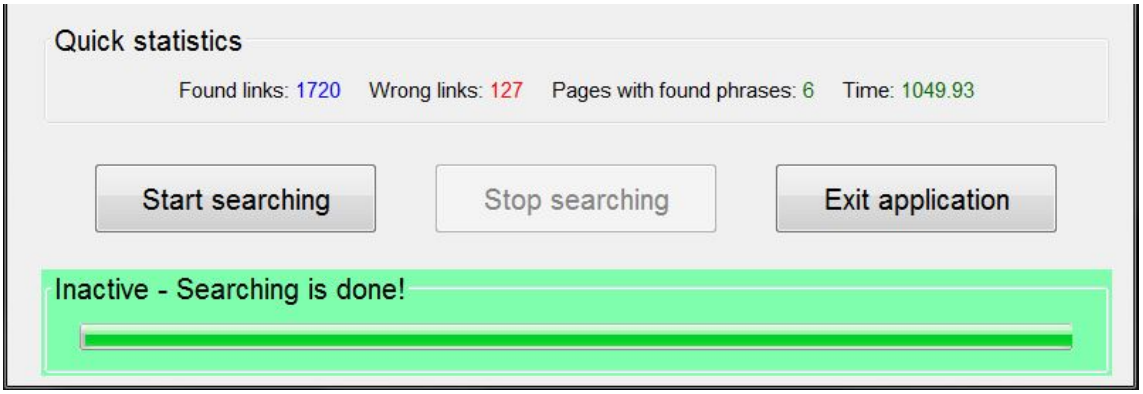
Poslední dvě komponenty tvoří tlačítka *Export*, které umožňuje uložit aktuální graf do *xml souboru* a tlačítko *zavřít (Close)*, kterým se celé toto okno zavře.

### 5.2.3 Ovládací panel

Tento panel je zobrazen na obrázku 13 a zůstává pořád na svém místě i při přepínání záložek. To umožňuje např. ukončit vyhledávání z kterékoliv záložky.

- **Rychlá statistka (Quick statistics)** - Na tomto místě je při vyhledávání ve stejném čase zobrazován přehled o následujících údajích.
  - **Nalezené odkazy (Found links)** - Zde je uživateli zobrazeno, kolik odkazů bylo nalezeno. Počet se obvykle navyšuje o jedničku, ale v některých případech se akce vyhledávání provede tak rychle, že se číslo zvětší o několik desítek.
  - **Chybné odkazy (Wrong links)** - Počet odkazů, které jsou z nějakého důvodu chybné. Důvody jsou specifikovány na záložce *chybový log (Error log)*.
  - **Počet odkazů s nalezenými frázemi (Pages with found phrases)** - Z názvu je poměrně jasné, že se jedná o počet stránek, na nichž se nachází minimálně jedno hledané slovo.
  - **Čas (Time)** - Čas vyhledávání který běží s přesností *100 milisekund*.

- **Ovládací prvky** - Další část tvoří tři hlavní tlačítka, pomocí nichž se *robot7* ovládá. Tlačítka jsou podle očekávaných akcí povolena či zakázaná. Tlačítka jsou následující.
  - **Spustit vyhledávání (Start searching)** - Tlačítko, kterým uživatel spouští vyhledávání, jsou-li všechny vstupní údaje v pořádku.
  - **Zastavit vyhledávání (Stop searching)** - Tímto tlačítkem se zastavuje vyhledávání. V některých případech může nastat situace, že při pokusu o zastavení *robot7* stálé poběží a nereaguje. To je ovšem na první pohled chybná domněnka. Je nutno počkat až se dokončí všechny potřebné akce a program se sám zastaví. Vyhledávání probíhá využíváním rekurzivního volání stejné funkce a díky tomu nemůže být *robot7* ukončen ihned, protože v *Javě* jsou mechanismy, které musí řádně ukončit rekurzivní volání a to nějaký čas trvá.
  - **Vypnout aplikaci (Exit application)** - Vypnutí celé aplikace.
- **Status bar** - Poslední části v tomto panelu je *Status bar* s popiskem, který indikuje aktuální status *robota7*. Celkem může nastat následujících pět stavů.
  - **Neaktivní (Inactive)** - Výchozí stav. *Status bar* je bez pohybu.
  - **Aktivní (Active)** - Stav, který se aktivuje po spuštění vyhledávání. *Status bar* je v pohybu a pozadí je zelené.
  - **Aktivní - Prosím počkejte. Musí být dokončeny poslední potřebné věci (Active - Please wait. It must be finish last needed things.)** - Tento stav nastane po stopnutí aplikace, než se adekvátně dokončí celý proces, z důvodu korektního ukončení rekurzivního volání. Pozadí při tomto stavu je zelené.
  - **Neaktivní - Vyhledávání bylo zastaveno (Inactive - Searching was stopped)** - Vyhledávání bylo uživatelem přerušeno. Pozadí je červené.
  - **Neaktivní - Vyhledávání bylo dokončeno (Inactive - Searching was done!)** - Vyhledávání bylo úspěšně dokončeno. Pozadí je zelené.




Obrázek 13: Hlavní ovládací panel

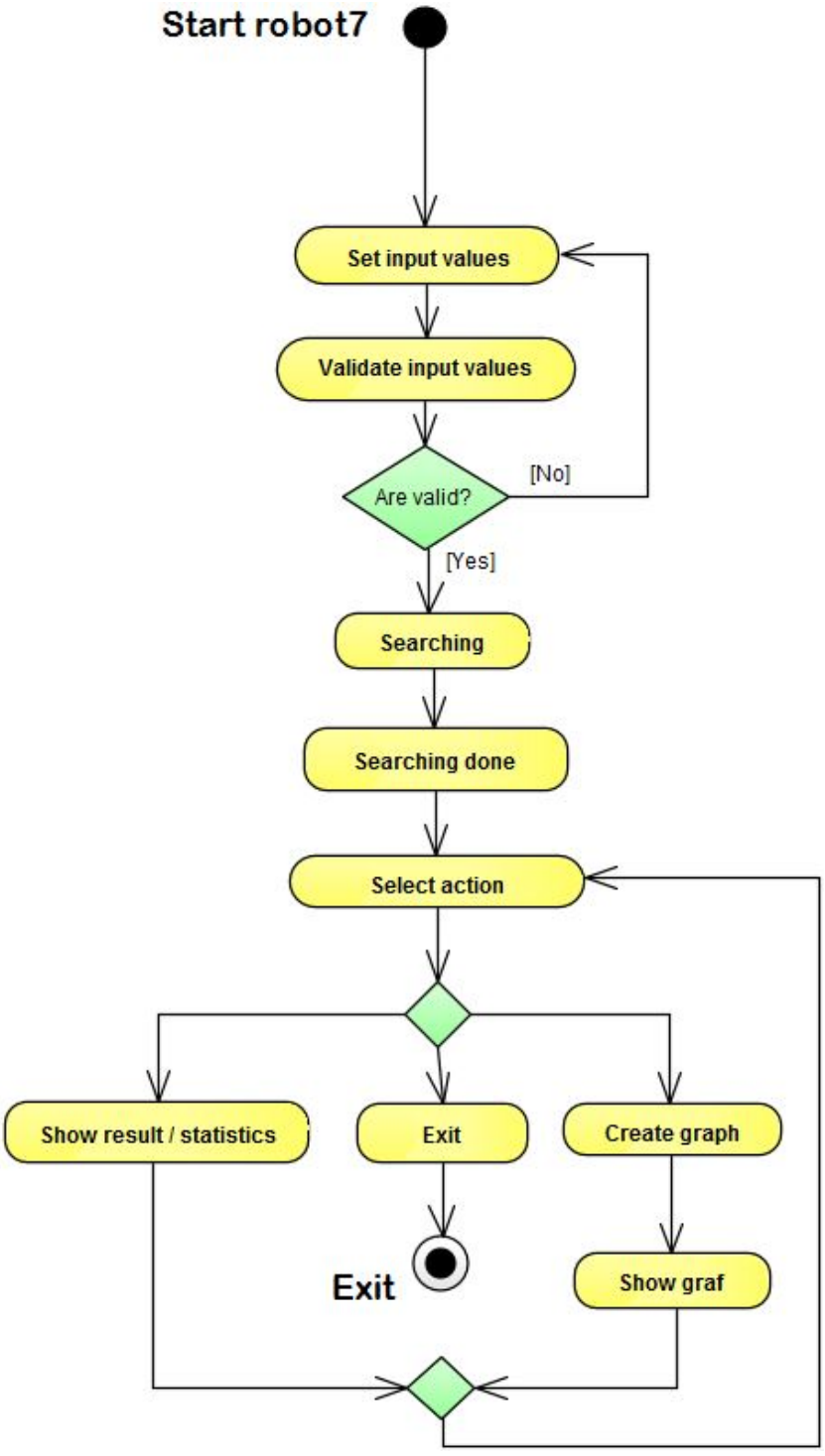
## 5.3 Vyhledávání

Jádrem celé aplikace je vyhledávání, a proto mu zde věnuji celou sekci. K názornějšímu pochopení použiji diagramy aktivit, u kterých jednotlivé aktivity podrobně popíši, jak *robot7* funguje na pozadí.

### 5.3.1 Celkový náhled

Aktivity diagram 14 zjednodušeně zobrazuje, jak *robot7* obecně funguje. Následně popíši jednotlivé aktivity.

- **Set input values** - První aktivitou je nastavení vstupních hodnot. Tím se rozumí minimálně zadání *defaultní URL adresy*, na které má začít vyhledávání. Ostatní hodnoty můžou zůstat ve výchozím nastavení a vyhledávání bude fungovat.
- **Validate input values** - Před spuštěním vyhledávání robot ještě jednou zkontroluje vstupní hodnoty, zda jsou korektní. *Robot7* je napsán tak, ať je možné do aplikace vložit pouze validní hodnoty, ale pokud uživatel edituje některý ze souboru *(maximumDepth.txt, maximumPages.txt, domains.txt)*, ze kterého se načítají hodnoty, tak přece jen k chybě může dojít a *robot7* bude uživatele varovat. Celý proces validace je detailně popsán v dalším diagramu 15. V případě, že validace neprojde, uživatel musí zadat nové hodnoty. V opačném případě přichází na řadu hlavní část *robota7* a to je vyhledávání.
- **Searching** - Aktivita vyhledávání je taktéž detailně popsána v samostatném diagramu 16.
- **Searching done** - Po ukončení vyhledávání je potřeba nastavit poslední důležité kroky. Stopnout běžící *timer*, zastavit *status bar* a nastavit *status*. Po dokončení si uživatel může vybrat, jak s výsledky naloží.
- **Show result/statistics** - Po ukončeném vyhledávání si může uživatel projít jednotlivé statistiky (počet nalezených odkazů, počet špatných odkazů, doba vyhledávání, log chyb, procházení stromu). Po dokončení této aktivity si uživatel může nechat vygenerovat graf nebo celou aplikaci ukončit.
- **Create graph** - Další možnosti po dokončení vyhledávání je vytvoření grafu na základě nalezených výsledků.
- **Show graph** - Po vytvoření grafu se uživateli zobrazí graf s řadou možností 5.2.2. Po dokončení této aktivity má uživatel možnost opět výběru, zda si bude procházet statistiky, vytvářet graf nebo celou aplikaci ukončí.
- **Exit** - Poslední možností po dokončení vyhledávání je ukončit celou aplikaci.

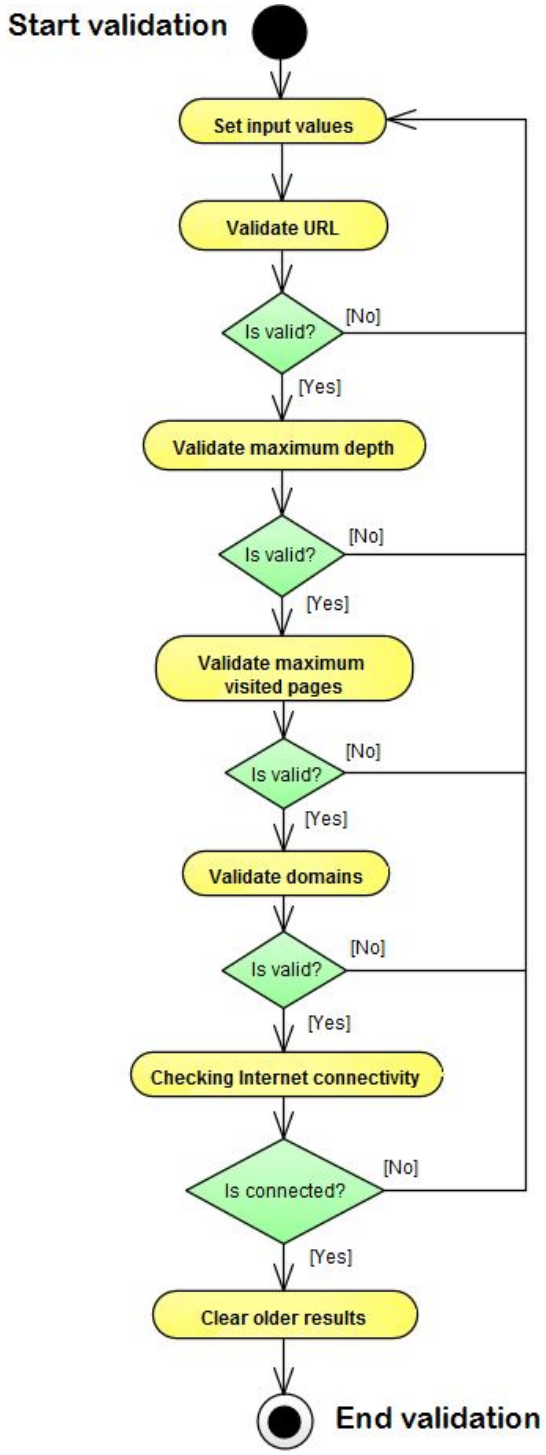



Obrázek 14: Aktivity diagram - celkový náhled

### 5.3.2 Validace vstupních hodnot

Další diagram zobrazuje proces validace, který se provede před spuštěním vyhledávání. Jedná se především o validaci hodnot, které uživatel zadá na úvodní záložce *(Search criteria)* a o kontrolu připojení k internetu. Pro všechny aktivity platí stejný scénář. Není-li hodnota validní, tak se celý proces vrací na začátek a uživatel musí zadat korektní hodnoty nebo se připojit k internetu.

- **Set input values** - Tato aktivita už byla popsána v předchozím aktivity diagramu 5.3.1.
- **Validate Url** - První validaci je kontrola *URL adresy*. Pro tuto validaci jsem použil validátor od *Apache* 5.1.2, který má v sobě sofistikovanou logiku na validaci. Dále je tu kontrola, jestli je hodnota vůbec zadaná.
- **Validate maximum depth** - U maximální možné hloubky zanoření se kontroluje zda je hodnota zadaná a druhým problémem může být načtení nekorektních hodnot ze souboru *maximumDepth.txt*. To se může stát, pokud uživatel do souboru vloží jiné hodnoty než čísla.
- **Validate maximum visited pages** - Maximální počet prohledaných stránek se validuje stejně jako předchozí hodnota *maximum depth* s tím rozdílem, že výchozí hodnoty se načítají z *maximumPages.txt*.
- **Validate domains** - Validace domény jen kontroluje, zda je hodnota zadaná. Pokud uživatel zadá doménu, která má špatný formát, tak se nic vážného nestane, ale stránky s jeho doménou nebudou nalezeny a výsledek bude pravděpodobně prázdný.
- **Checking Internet connectivity** - Tato kontrola zjišťuje, zda je počítač, na kterém je spuštěn *robot7*, připojen k internetu. Při tomto zjišťování jsem použil několik známých serverů, od kterých se snažím získat odpověď. Pokud odpověď nedostanu ani od jednoho serveru, tak počítač není pravděpodobně připojen k internetu. Serverů na dotazování používám více, protože se může stát, že některý bude mít výpadek a došlo by k mylnému rozhodnutí při prvním dotazu. Mnou testované servery jsou: *google.com, amazon.com, seznam.cz, centrum.cz, youtube.com a twitter.com*.
- **Clear older results** - Jsou-li všechny validace úspěšné, tak je zde potřeba připravit *robota7* na nové vyhledávání. To obnáší smazání starých statistik, vymazání logu a nastavení *statusu*.




Obrázek 15: Aktivity diagram - validace vstupních hodnot

### 5.3.3 Vyhledávání

Nejpodstatnější část robota, kterého jsem naprogramoval, se nachází právě v této aktivitě. Je zde přehledně znázorněna posloupnost jednotlivých kroků, které se při vyhledávání provádí. Vstupem každého vyhledávání je *URL adresa*, na které se provádí veškeré operace.

- **Check interrupt** - Před dotazem na konkrétní *URL adresu* je kontrola, zda nebyl robot uživatelem zastaven. To se provede tlačítkem *Stop searching*. Byl-li zastaven, tak *robot7* ukončí své vyhledávání a celý proces je ukončen.
- **Fixing URL** - V této aktivitě se provádí s *URL adresou* poslední úpravy potřebné pro dotaz na server. Například se zde nahrazují zpětné lomítka za normální a jiné potřebné úpravy spojené především s požadavky programátorskými *API*. Pokud je upravena *URL adresa* nevalidní, tak vyhledávání pokračuje aktivitou *Check pool*, která je dále popsána. Proč se ale kontroluje *URL adresa* ještě na tomto místě, když už je kontrolována jednou na místě, kde se zadává vstupní *URL adresa*? Je to proto, protože zde už pak přichází nalezené odkazy a ty nejsou přece ty zadávané na vstupu, aby mohly být validovány. V případě úplně první *URL adresy* se tato kontrola provede ještě jednou a *robota7* to nijak nezpomalí.
- **Check delay** - Je-li nastaven *timeout*, tak se provede krok *Process delay*, jinak přichází na řadu krok *Connect to URL*
- **Process delay** - Provede se zpoždění *robota7* a přichází na řadu krok *Connect to URL*.
- **Connect to URL** - Na tomto místě se stahuje konkrétní stránka. Stahování provádím pomocí *Jsoup API* 5.1.2
- **Checking searched words** - Dále je potřeba zkontrolovat, zda uživatel hledá nějaké slova na stránce. To znamená, že zadal nějaké slova do panelu *Searched words* na záložce *Search criteria*. Pokud tedy uživatel něco zadal, *robot7* odbočí do větve, kde celou staženou stránku zpracuje. Pokud uživatel žádné slova nevyhledává, tak přechází do aktivity *Check input conditions*.
- **Clear html file** - Tato aktivita je první v pořadí, pokud uživatel požaduje vyhledat nějaké slovo. Zde začíná mnou odlehčený vyhledávač slov, který začíná právě touto aktivitou. Píši odlehčený, protože oproti známým vyhledávačům je vyhledávání odlehčené. Právě na tomto místě se provádí vyhodnocování a indexování stránek, což pak určuje šanci ve výsledku vyhledávání. Pro vyhledávání jsem zvolil následující postup.
  Přímo v této aktivitě se provádí vyčištění *HTML stránky*, čímž je myšleno získání pouze *HTML tagů* a jejich obsahu. Za touto aktivitou následuje aktivita *Escape html tags*.
- **Escape html tags** - Zde se z výsledku předchozí aktivity odstraňují *HTML tagy* a je získán pouze čistý obsah (slova). Na celém textu je zde provedena ještě změna

všech velkých písmen na malé. Tímto nastavují, že vyhledávání není *case sensitive*. Následuje *Create dictionary*.

- **Create dictionary** - Zde získány čistý text rozdělím podle mezer na jednotlivé slova a tyto slova uložím do kolekce. Zde mi ale nastal problém, protože do kolekce jsou uloženy slova i s nežádoucími znaky. Protože jsem rozdělil slova po mezerách. Problém vysvětlím na následující aktivitě.

- **Fix dictionary** - Zde se provádí úpravy nad vytvořeným slovníkem. Představme si následující příklad, kdy je na stránce následující věta.

  *Okolo 10-té hodiny začíná 2. maraton, nyní na téma "Clean".*

  Představme si, že budu vyhledávat slovo *kolo* a udělal bych to nejjednodušším možným způsobem a to, zda získaný text ze stránky obsahuje textový řetězec *kolo*. Odpověď by byla pravdivá, protože mnou hledané slovo je součástí slova *Okolo* a takové chování je nežádoucí. Proto jsem zvolil následující techniku.
  Aktivita *Create dictionary* mi z uvedené věty vytvoří následující množinu slov, které jsou vloženy do kolekce typu *Set*. Do kolekce typu *Set* proto, protože mému robotu stačí pouze jeden výskyt slova na stránce. Původně jsem používal kolekci *List*, ale dotaz nad kolekci *Set* je rychlejší a celkový počet slov vkládaných do kolekce byl při testování o dvě třetiny menší, protože na stránkách se slova často opakují.

  {*okolo, 10-té, hodiny, začíná, 2., maraton, nyní, na, téma, "clean".* }

  Nyní mohu provést kontrolu nad kolekcí, zda hledané slovo obsahuje a výsledek už bude korektní a to negativní. Ovšem, co když budu hledat slovo *clean*? *Robot7* by opět nefungoval správně, protože slovo *clean* je ve slovníku uvedeno mezi uvozovkami a porovnání by nebylo korektní. Proto je nutné udělat další krok, aby byl výsledek korektní.
  Nad touto množinou jsou v *robotovi7* procházeny jednotlivé slova a pokud obsahují nějaký speciální znak, tak je nahrazen prázdným znakem a výsledné slovo je přidáno do původní kolekce. Tím vznikne nová kolekce, se kterou už se bude pracovat lépe. Definované speciální znaky v *robotovi7* jsou uvedeny v následující množině.

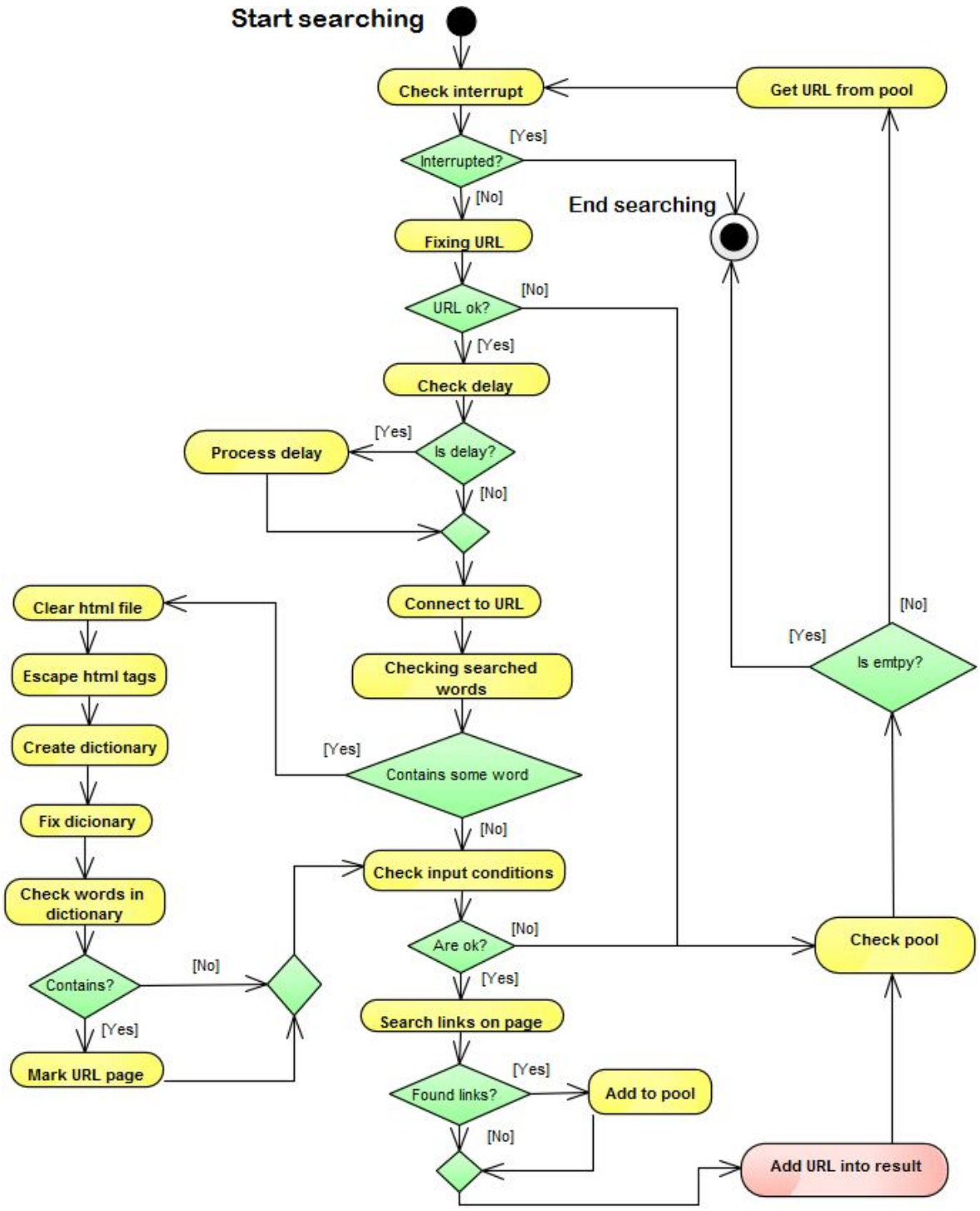
  {!, ", #, $ , %, &, ', (, ), *, +, -, ., /, :, ;, <, >, ?, @, [ , \, ], ^, _, ›, {, |, }, ~, „, “}

  A zde je nově vytvořena množina.

  {*okolo, 10-té, hodiny, začíná, 2., maraton, nyní, na, téma, "clean"., 10, té, hodiny, začíná, 2, maraton, nyní, clean* }

  Bude-li se nad touto množinou hledat slovo *clean*, tak výsledek už bude správný a to kladný. Zde je nutnost vyhledávanému slovu nastavit všechny písmena malé.

- **Check words in dictionary** - V této části se provede již zmíněný dotaz, zda se slovo nachází ve slovníku (stránce). Pokud ano, tak se provede aktivita *Mark URL page*. V opačném případě je provedena aktivita *Check input conditions*.
- **Mark URL page** - Procházená stránka obsahuje hledané slovo a tak se zde tento fakt zaznamená a pokračuje se v procesu dále a to aktivitou *Check input conditions*.
- **Check input conditions** - Zde se provádí kontrola, zda se stránka nenachází v zakázané hloubce, nepřekročil se maximální počet stránek a obsahuje specifikované domény. Je-li vše v pořádku přejde se k aktivitě *Search links on page*. V opačném případě se přechází k aktivitě *Check pool*.
- **Search links on page** - V tomto kroku se na stránce hledají odkazy a jsou-li nějaké nalezeny, tak je zde přechod k aktivitě *Add to pool*. V opačném případě je následujícím krokem aktivita *Add URL into result*
- **Add to pool** - V tomto kroku se nalezené odkazy přidávají do poolu pro následné vyhledávání. Poté se přichází na řadu aktivita *Add URL into result*.
- **Add URL into result** - Celý proces vyhledávání pro aktuální stránku prošel v pořádku a zde je přidán do výsledku (Strom odkazů na záložce *Result tree*). Následujícím krokem je aktivita *Check pool*.
- **Check pool** - Na tomto místě se zkontroluje, zda jsou k dispozici ještě nějaké odkazy, pro následující vyhledávání. Pokud ano, tak přichází na řadu aktivita *Get URL from pool*. V opačném případě *robot7* ukončuje svou činnost.
- **Get URL from pool** - Na tomto místě se vybere nalezený odkaz a celý proces začíná znovu na aktivitě *Check interrupt*. Při vyhledávání používám rekurzivní volání, takže v tomto případě je *poolem* zásobník.




Obrázek 16: Activity diagram - vyhledávání

## 5.4 Testování

Při testování jsem očekával, že budu velmi často jednotlivými servery při jejich prohledávaní blokován, proto jsem v *robotovi7* zohlednil i možnost nastavení *timeoutu* mezi jednotlivými dotazy. Ve skutečnosti jsem po celou dobu testování nenarazil na stránku, která by mě zablokovala nebo zabránila stahovat obsah stránek. Tento fakt byl pro mě překvapující, protože prohledávání neustále běží a stahuje obsahy stránek. Navíc soubor *robots.txt* není brán v potaz. Servery tedy neberou mého robota jako hrozbu nebo jsou mizerně zabezpečené.

### 5.4.1 Verze programů

Zde jsou uvedeny všechny verze jazyka a knihoven, které byli potřebné pro vývoj.

- **Java** - Build-Jdk: 1.7.0_55
- **Maven** - 3.2.1
- **Jsoup** - 1.8.1
- **Jung** - 2.0.1
- **Apache commons validator** - 1.4.0

### 5.4.2 Kompatibilita operačních systémů

*Robot7* je naprogramován v jazyce *Java*, který je z pohledu operačních systému multiplatformní. To je sice pravda, ale každý operační systém má pro své grafické rozhraní jinak nastylované komponenty, které se můžou ve výsledku lehce lišit. *Robot7* funguje minimálně na následujících dvou operačních systémech, na kterých byl řádně otestován.

- **Windows 7 Enterprise** - 64-bit
- **Linux Ubuntu** - 14.04 LTS

### 5.4.3 Testovací zařízení

V *robotovi7* uvádím i čas vyhledávání, který je potřeba na dokončení celého procesu. Může tedy hrát roli na jak výkonném zařízení je *robot7* spouštěn a jak rychlé má uživatel připojení k internetu, proto je uvedena následující specifikace, na které je závislý čas celého procesu.

- **Operační systém** - Windows 7 Enterprise 64-bit
- **Procesor** - Intel Core i7 4800 MQ
- **Operační paměť** - 16 GB DDR3
- **Grafika** - Intel HD Graphics 4600

### 5.4.4 Tabulky

V následující tabulce jsou údaje o vyhledávání s různým nastavením. Prvně uvedu *URL adresy* a přiřadím jim čísla, pod kterými jsou tyto *URL adresy* prezentovány.

- **1** - `http://www.vsb.cz/`
- **2** - `http://www.cs.vsb.cz/`
- **3** - `http://www.google.com`

| Default URL | Maximum depth | Maximum visited pages | Only default domain | Searched words | Timeout | Found links | Wrong links | Pages with found phrases | Time (s) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | unlimited | unlimited | yes | sojka | no | 21694 | 471 | 110 | 7298.11 |
| 1 | unlimited | unlimited | yes | zelinka | no | 21882 | 418 | 30 | 6375.94 |
| 1 | 3 | unlimited | yes | informatika | no | 11751 | 740 | 116 | 7928.86 |
| 1 | 3 | unlimited | yes | informatika | no | 1487 | 34 | 25 | 413.89 |
| 1 | unlimited | 1500 | yes | informatika | no | 1500 | 137 | 1 | 463.26 |
| 1 | unlimited | 1500 | yes | | no | 1500 | 139 | 0 | 576.89 |
| 1 | 3 | unlimited | yes | | no | 1487 | 34 | 0 | 367.94 |
| 2 | 3 | unlimited | yes | informatika | no | 2892 | 167 | 107 | 1341.06 |
| 3 | 1 | unlimited | yes | | no | 22 | 4 | 0 | 5.67 |
| 3 | 2 | unlimited | yes | | no | 657 | 22 | 0 | 250.96 |
| 3 | 3 | unlimited | yes | | no | 4741 | 473 | 0 | 2146.22 |

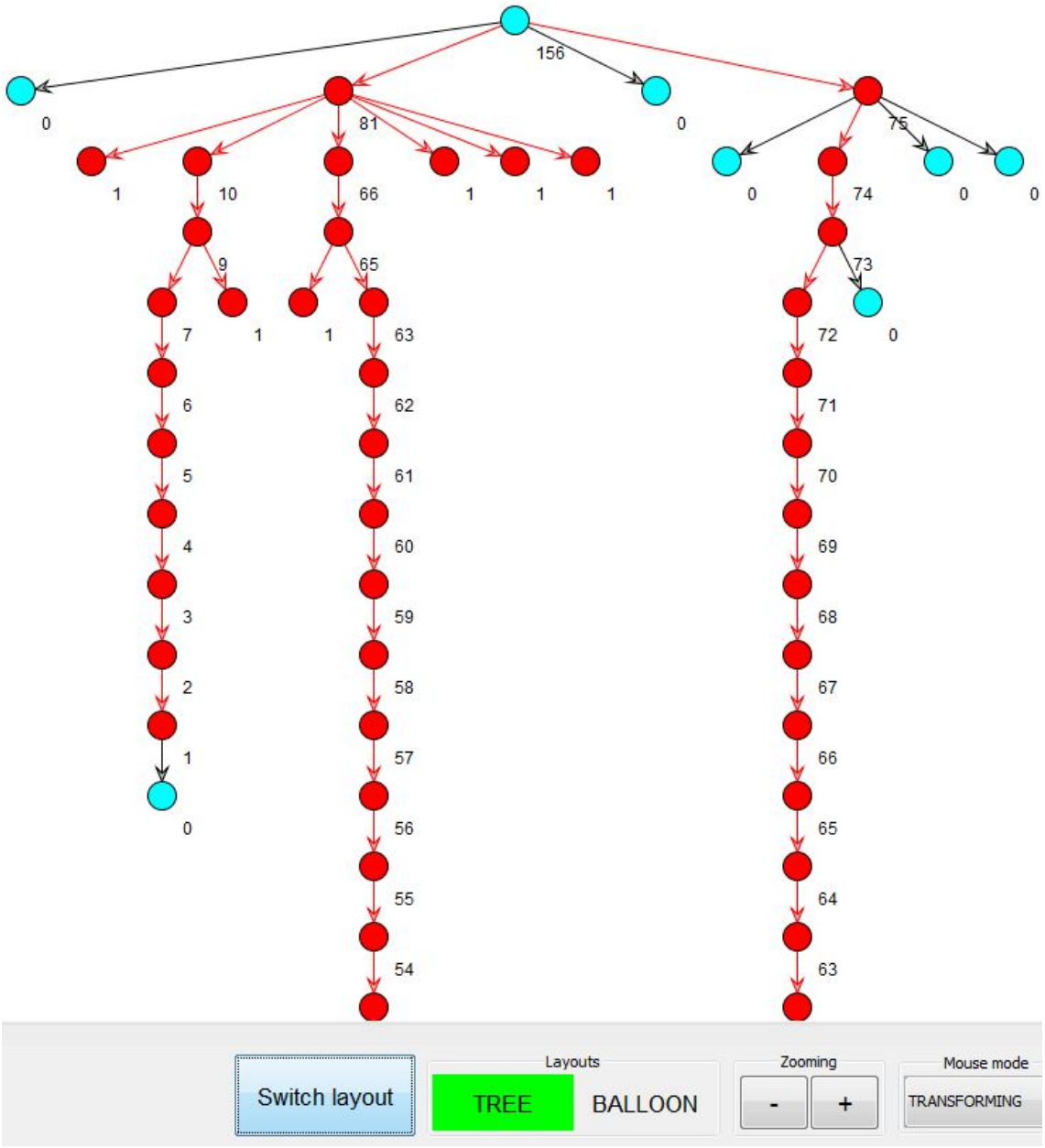
Tabulka 1: Výsledky vyhledávání

Na uvedené tabulce 1 jsou vidět zajímavé statistické údaje. Přímo na stránce *www.google.com*, což je přímo vyhledávač, bylo nalezeno 22 odkazů během 5 sekund. O úroveň níže už těchto odkazů bylo nalezeno 657. Na třetí úrovní stejné stránky už bylo odkazů 4741 a toto číslo se s narůstajícími úrovněmi velmi hodně navyšuje. Z jednoho odkazu je tedy poměrně snadno získáno mnoho dalších.

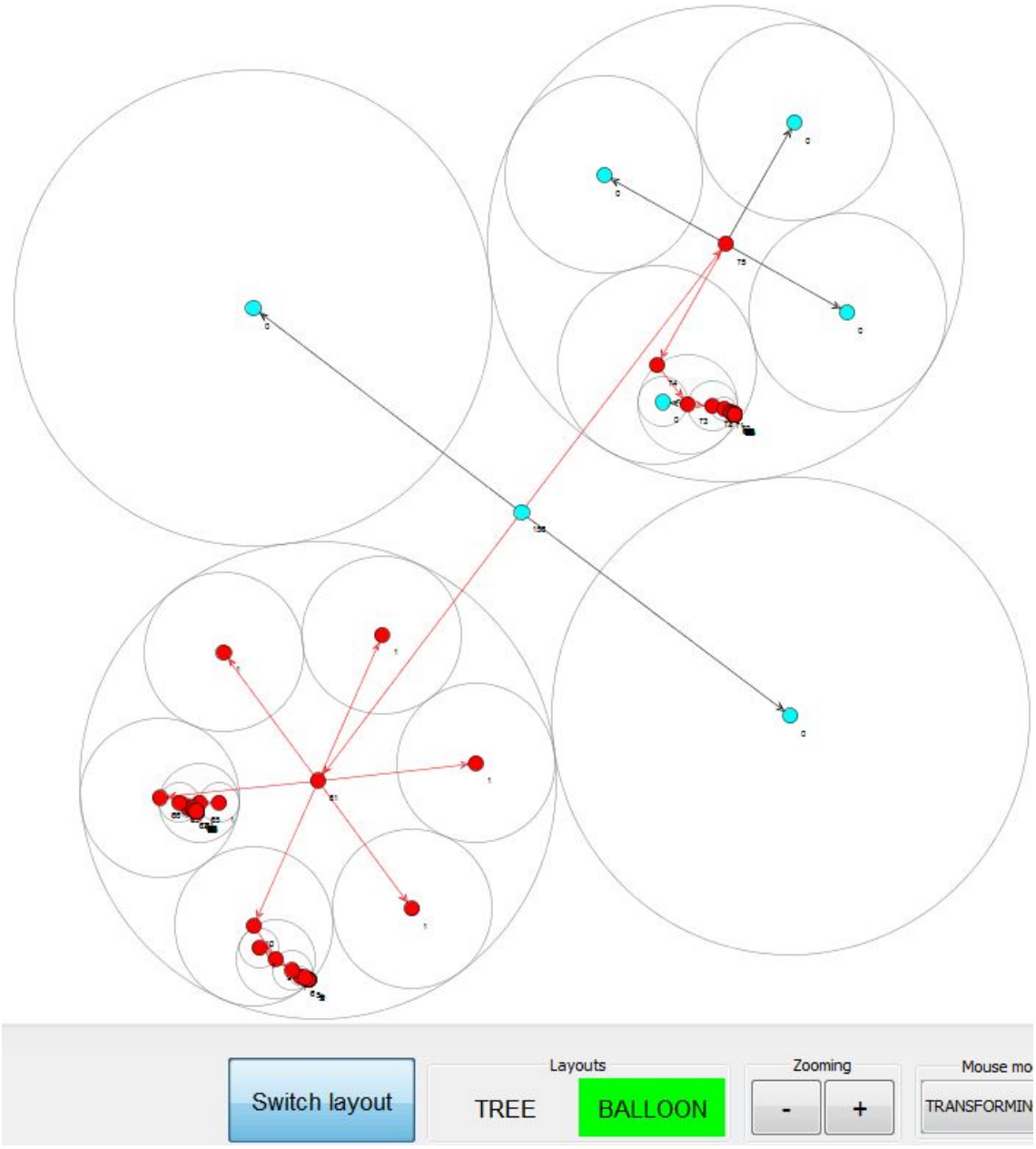
*Robota7* je vhodné pro konkrétní vyhledávání dobře nastavit, aby vyhledával požadované údaje. V případě výchozího nastavení je vyhledávání téměř nekonečné. *Robota7* lze taky velmi dobře využít pro detekci odkazů, na neexistující stránky. Tyto stránky lze snadno vyčíst z *Error logu* a administrátorům serverů by tato funkcionalita byla jistě užitečná. Odkazů na neexistující stránky je na internetu skutečně mnoho, jak jsem v tomto testování zjistil.

### 5.4.5 Vytvořené grafy

Vytvářené grafy v *robotovi7* velmi názorně ukazují, jak jsou stránky propojeny. Na obrázku 9 je vidět dokonalé využití rozvržení *balloon*. Zde ještě ukážu, že i rozvržení *tree* má význam v některých případech. Obvykle se jedná o velké zanoření na pár větvích. Obrázek 17 to názorně dokazuje. Ten samý graf v rozvžení *balloon* nepůsobí zajímavě 18.




Obrázek 17: Graf - v některých případech má využití i rozvržení *tree*




Obrázek 18: Graf - stejný graf jako 17 v rozvržení *balloon*

### 5.4.6 TOR

Zajímavým pokusem bylo pokusit spustit *robota7* přes anonymizující síť *TOR*, aby vyhledávání *robotem7* probíhalo anonymně. V minulosti *TOR* nabízel plugin do prohlížeče *(Mozilla firefox)* a po jeho aktivaci začala *TOR* anonymizace fungovat v tomto prohlížeči. Dnes už *TOR* nabízí ke stažení následující dva balíčky, které jsem stahl pro operační systém *Windows*.

- **Tor Browser** - Tento instalační balíček je velký *32,8 MB*. Nainstaluje se tím prohlížeč, ve kterém je *TOR* zakomponovaný a anonymizace přes něho skutečně funguje. Bohužel funguje jenom v tomto prohlížeči a při spuštěném *TOR browseru robot7* anonymně neprohledává.
- **Expert Bundle** - Tento balíček o velikosti *3,2 MB* obsahuje jen sadu knihoven, které tvoří *TOR*. Konkrétní aplikace se pak musí ručně nakonfigurovat, aby byly schopné *TOR* využívat. Bohužel tuto možnost nezohledňuji v *robotovi7*, ale existuje tedy možné řešení, jak *robota7* v budoucnu vylepšit.

# 6 Implementace vlastní ochrany před roboty

Součástí této práce má být také implementace ochrany před monitorováním *digitální stopy*. V kapitole 4.2 už jsem uvedl možné způsoby detekování internetových robotů a právě z toho budu dále vycházet. Je-li server špatně zabezpečen na straně síťových zařízení, tak můžou roboti server zahlcovat dotazy a získávat obsah stránek, který dále zpracovávají dle potřeby.

Navrhl jsem tedy řešení přímo na aplikačním serveru, kdy mám serverovou aplikaci, na které mám řadu různě propojených internetových stránek. Principem je, že před každým klientským požadavkem bude provedena určitá kontrola, která se pokusí na základě nějakých pravidel rozhodnout, zda se jedná o robota či nikoli. Toto řešení sice klade větší nároky na straně aplikačního serveru a může způsobit menší časovou prodlevu, ale je potřeba zvážit, zda je bezpečnost důležitější, než načtení stránky např. s půl sekundovým zpožděním. Pro tuto aplikaci je dále použit název *servlet-filter*.

## 6.1 Použité technologie

Všechny technologie zde použité jsou zcela zdarma a programoval jsem opět v jazyce *Java*. Výsledkem této implementace je *webová aplikace*, kterou jsem pojmenoval *servlet-filter*. Tato aplikace obsahuje sadu propojených stránek a jeden *servlet*, který provádí detekci robota. *Servlet* je program naprogramovaný v jazyce *Java*, který zpracovává *HTTP požadavky* a generuje *HTML stránky*. Výsledný soubor se nazýva *servlet-filter.war*.

### 6.1.1 Java

Zde jsem použil stejný programovací jazyk jako v předchozí kapitole 5.1.1. Důležitá zde byla možnost napsat nějakou aplikaci, u které bude možné definovat místo, kde se provede detekce robota. To umožňují právě *servlety*, kde je možnost nastavení filtrů, které se provádí před načtením stránky. Nastavení filtrů se provádí v souboru *web.xml*, který je součástí *WAR archivu*.

### 6.1.2 Apache Tomcat

Aplikaci jsem testoval na aplikačním serveru *Apache Tomcat* [43], který podporuje deploy souborů typu *WAR*.

## 6.2 Detekce robota

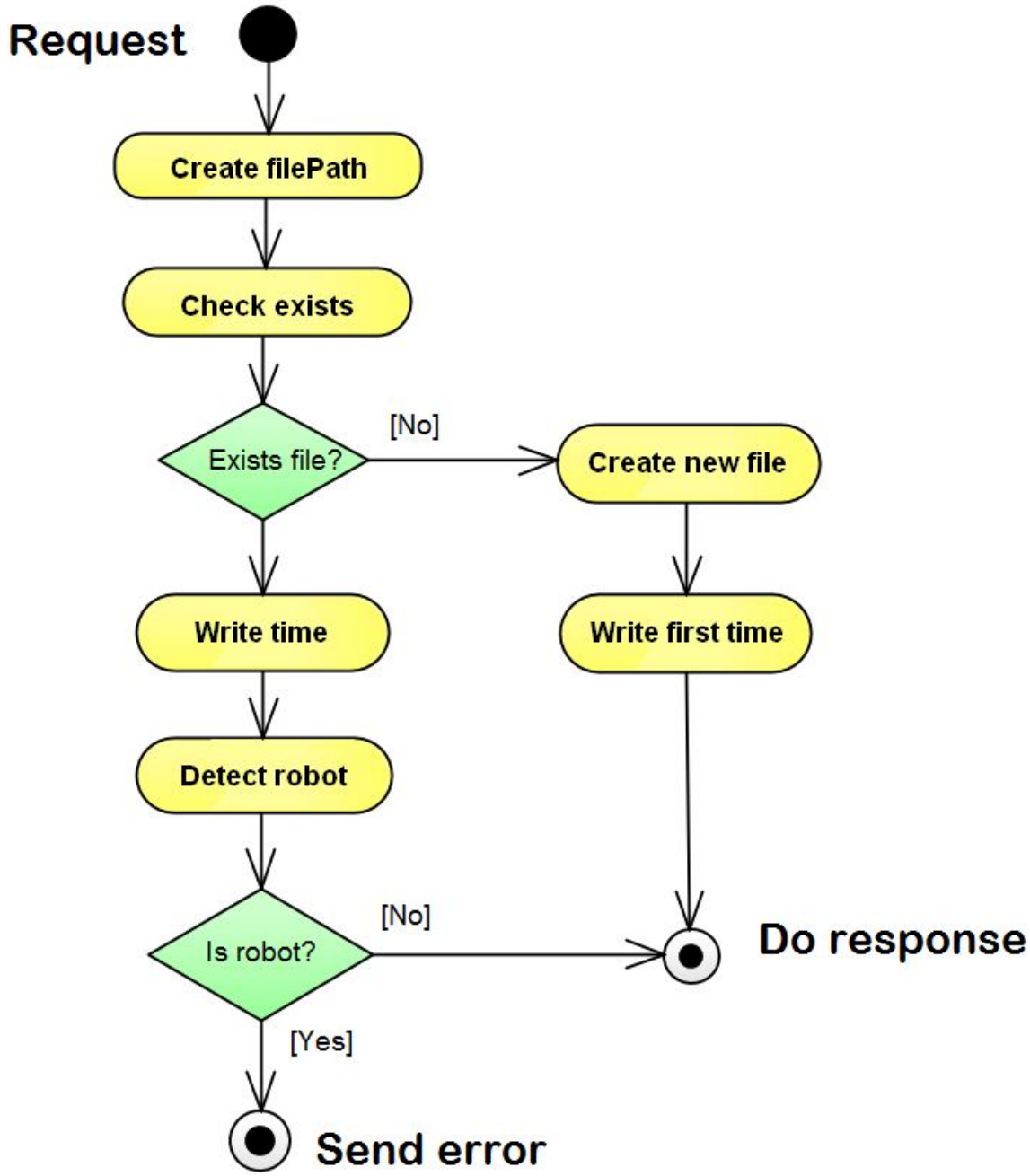
Detekce robota je popsaná na následujícím diagramu 19, na kterém jsou jednotlivé aktivity podrobně popsány. Detekce začíná požadavkem každého klientského dotazu na server.

- **Create filePath** - Na začátku každého požadavku se vytvoří cesta k souboru, kde jsou ukládány požadované informace potřebné k detekci. V případě *Apache Tomcatu* je to v kořenovém adresáři složka *temp*. Název souboru je vytvořen z *klientské*

*IP adresy*. Informace jsou ukládány do textového souboru a výsledná cesta může vypadat např. takto: *"kořenový adresář serveru + \temp\128.45.5.15.txt"*.

- **Check exists** - V tomto kroku se kontroluje, jestli existuje cesta k souboru, jinak řečeno, zda soubor existuje. Pokud neexistuje pokračuje se aktivitou *Create new file*. V opačném případě se pokračuje aktivitou *Write time*.
- **Create new file** - Touto aktivitou se vytvoří soubor specifický pro každou jinou *IP adresu*. Při každém požadavku z konkrétní *IP adresy* je do souboru uložen aktuální čas serveru v milisekundách. Každý další čas je na novém řádku.
- **Write first time** - Po vytvoření nového souboru je do něj zapsán první čas. Tímto je zřejmé, že uživatel přišel na server poprvé a uživateli je vrácena korektní odpověď (internetová stránka).
- **Write time** - Touto aktivitou je do již existujícího souboru zapsán aktuální čas serveru a následuje aktivita *Detect robot*.
- **Detect robot** - Zde se nachází hlavní logika detekce robota. Pro jeho detekci jsem navrhl dvě různé kritéria.
  - **Rychlost dotazů** - V tomto kritériu je vyhodnoceno posledních pět dotazů (posledních pět řádků ve specifickém souboru pro danou *IP adresu*) na základě jejich časové prodlevy. Kontrola spočívá v rozdílu mezi jednotlivými dotazy. Je-li mezi každým předcházejícím dotazem časová prodleva menší jak *100 milisekund*, tak tento klient je vyhodnocen jako robot, protože člověk by jen těžko mohl takto rychle proklikat jednotlivé stránky.
  - **Periodicita dotazů** - Tato metoda na detekování už je poměrně sofistikovanější než je v případě rychlosti dotazů. Zde se snažím detekovat roboty, kteří posílají dotazy na server v konstantních periodách. Zde je zohledňováno taktéž posledních pět dotazů a princip je následující.
    Z posledních dvou dotazů je určen rozdíl mezi dotazy. Tento rozdíl je nastaven jako *hlavní perioda* pro následující porovnávaní. Následuje určení nové periody z předposledního dotazu a jemu předcházejícímu dotazu. Opět je určen rozdíl mezi těmito dotazy a tento rozdíl je nastaven jako *nová perioda*. Z této *nové periody* je určen interval přičtením a odečtením *100 milisekund*. Nachází-li se *hlavní perioda* v tomto intervalu, tak je klient opět vyhodnocen jako robot.
    Vypočtený interval pro finální rozhodnutí je zde nutností, protože i když robot provádí periodické dotazy, tak vzhledem k latenci sítě je prakticky nemožné, aby se v rámci milisekund pětkrát po sobě dotazoval se stejnou periodou.

Je-li v některém ze dvou případů klient detekován jako robot, tak je mu vrácena chyba v podobě neexistující stránky (*HTTP 404*) a robotovi je zabráněn přístup k obsahu stránek. V opačném případě je uživateli vrácen korektní požadavek (stránka).




Obrázek 19: Aktivity diagram - detekce robota

## 6.3 Testování

Testování jsem prováděl na *localhostu* a následně je popsán celý průběh. Celé testování jsem prováděl na stejném zařízení jako v případě testování *robota7* 5.4.3. Pro otestování je třeba mít nainstalovaný aplikační server. V mém případě je to *Apache Tomcat*.

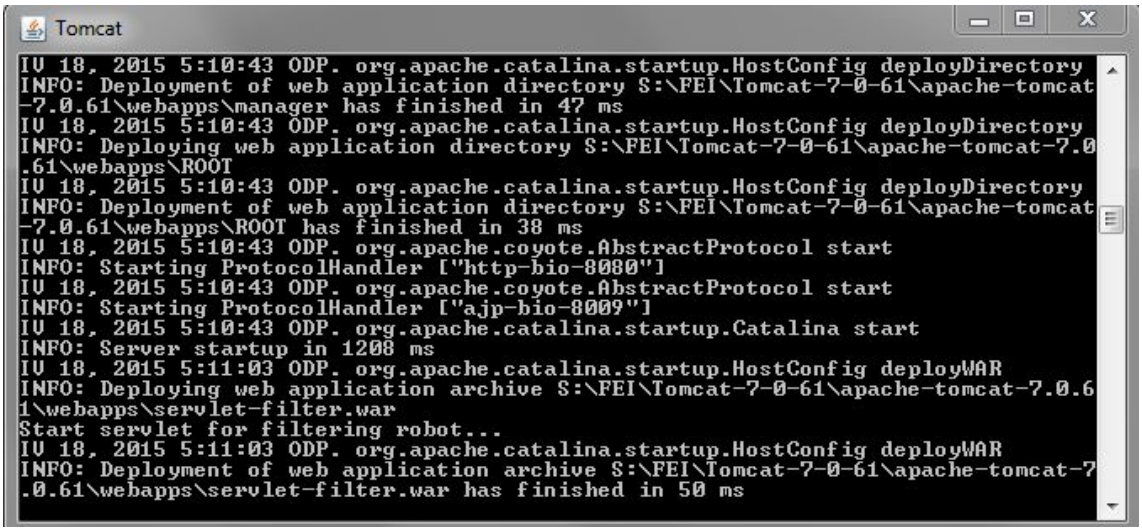
### 6.3.1 Verze programů

Verze programu v tomto testování jsou stejné jako v kapitole 5.4. Zde byl potřeba navíc akorát aplikační server *Tomcat*.

- **Apache Tomcat** - 7.0.61

### 6.3.2 Postup

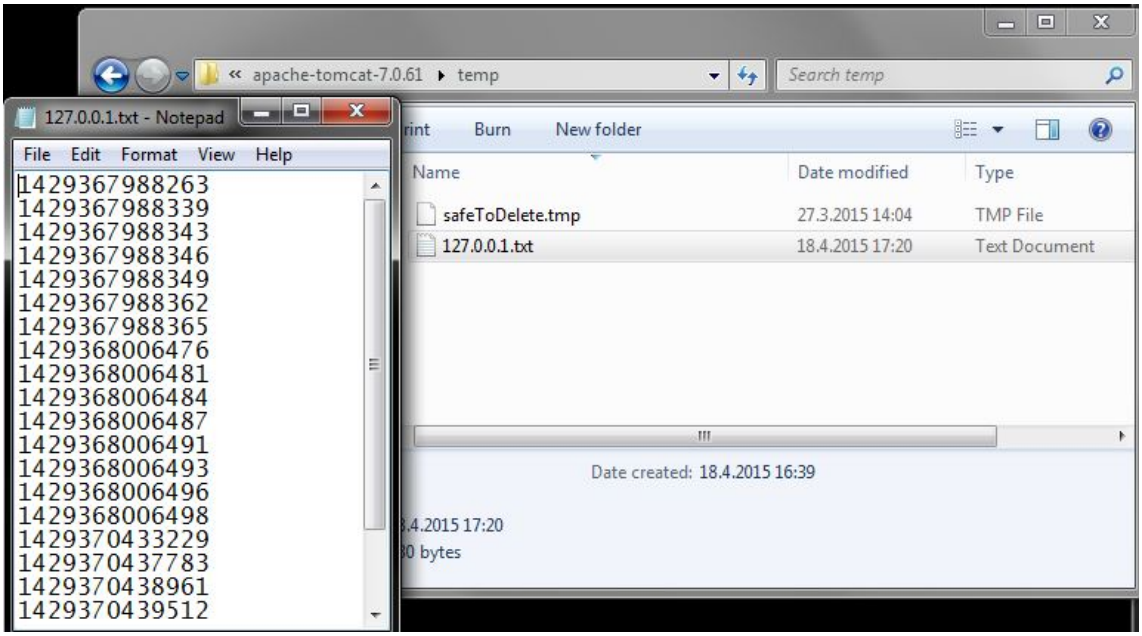
1. **Spuštění** - *Tomcat* spustím příkazem *startup.bat*. V jeho kořenovém adresáři se nachází složka *webapps*, do které nakopíruji vytvořenou aplikaci *servlet-filter.war*. Na obrázku 20.




Obrázek 20: Log *Tomcatu* - deploy *servlet-filter.war*

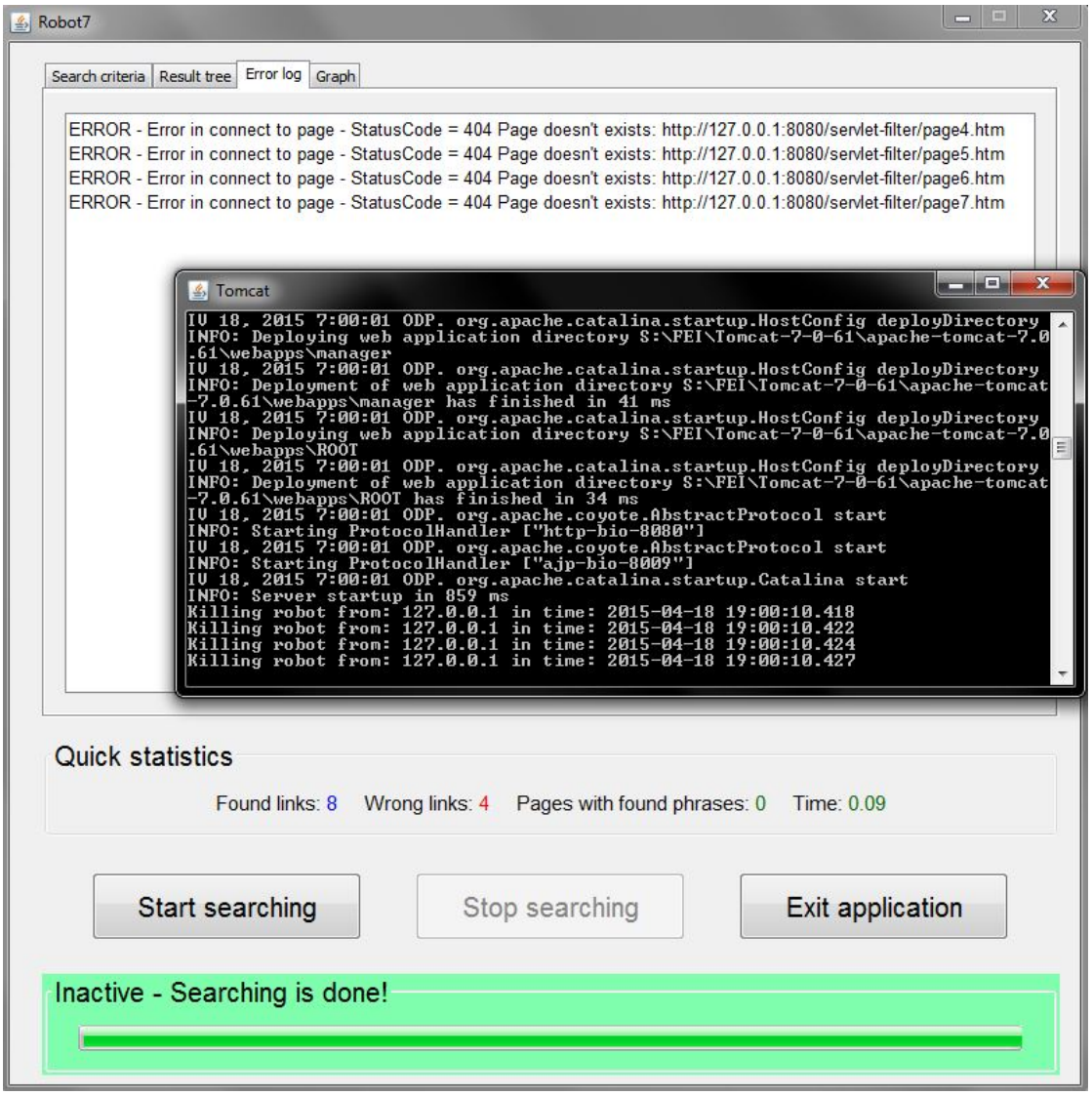
2. **Testování v prohlížeči** - Otevřu si prohlížeč a zkouším se proklikávat jednotlivými odkazy, které vedou na stránky v aplikaci *servlet- filter.war*. *URL* aplikace je *http://127.0.0.1:8080/servlet-filter*, kde *127.0.0.1* je adresa *localhostu*. Stránku v prohlížeči lze vidět na následujícím obrázku 21. Při proklikávání nenastal žádný problém a vždy byly všechny stránky korektně načteny.
   V adresáři */apache-tomcat-7.0.61/temp* byl vytvořen soubor *127.0.0.1.txt* a jeho obsah je vidět na obrázku 22


Obrázek 21: Aplikace *servlet-filter* v prohlížeči




Obrázek 22: Log klientského dotazu z IP adresy *127.0.0.1*

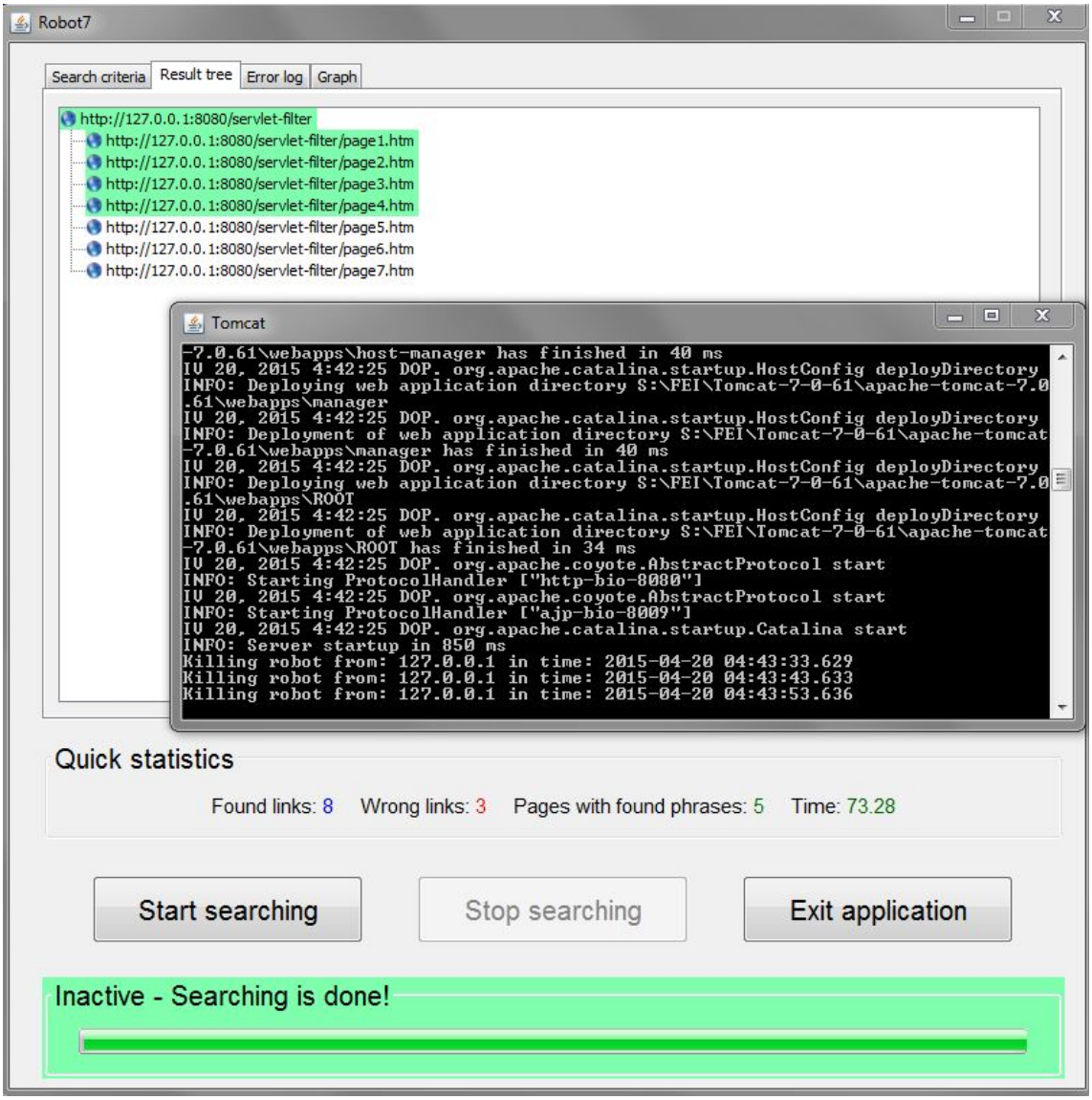
3. **Útok robotem (výchozí nastavení)** - Nyní přišel čas na *robota7*. Spustím jej tedy a jako výchozí *URL adresu* vložím *http://127.0.0.1:8080/servlet-filter*. Ponechám výchozí hodnoty a spustím prohledávaní. S *homepage* stránkou je k dispozici celkem osm stránek a v případě prvního vyhledávání je nalezeno osm odkazů a již pátý odkaz na stránku je detekován jako robot. Na následující tři odkazy již nejsou stránky ze serveru vůbec vráceny, tak jako při pátem dotazu. Na obrázku 23 je vidět, jak server odmítl robota. Časové rozestupy klientských dotazů jsou *2-4 milisekundy*, což je o mnoho méně, než nastavených *100 milisekund*.
   Dále je na tomto obrázku vidět, jak se s tím *robot7* vypořádal. Našel osm odkazů a z toho jsou čtyři odkazy špatné. Záznam o chybách je vidět na záložce *Error log*.




Obrázek 23: Ukázka rozpoznaného robota při velice rychlých požadavcích

4. **Útok robotem (použití timeoutu)** - Nastavení *robota7* bude, totožné jako v předchozím případě, ale v tomto pokusu jsem navíc zaškrtl položku *Set timeout for search? (ms)* a nastavil jsem časovou prodlevu deset sekund (10000 milisekund).
   Výsledek je možné vidět na obrázku 24. Po pěti prohledaných stránkách robot narazil a zbylé tři stránky nemohl prohledat. Tentokrát to je možné poznat ze záložky *Result tree*, protože jsem hledat slovo *some*, které je na všech osmi stránkách a poslední tři stránky jsou ve stromu bez zeleného pozadí. To signalizuje, že slovo nalezeno nebylo a server po detekci periodicity jednotlivé stránky neposlal. Z logu aplikačního serveru je vidět, že dotazy byly skutečně téměř periodické. Rozdíly jsou *3-4 milisekundy*.

5. **Výsledek** - V obou případech byl můj robot serverem rozpoznán a obsah stránek klientovi neposlal. Tím se server brání před roboty a monitorováním *digitálních stop*. Časové rozestupy *100 milisekund*, jež jsou v aplikaci *servlet-filter* použité, se můžou jevit jako zbytečně vysoké podle výsledku, ale to je především tím, že obě aplikace běží na jednom zařízení.




Obrázek 24: Ukázka rozpoznaného robota při periodických požadavcích

# 7 Závěr

Cílem této práce bylo poskytnout přehlednou a důkladnou studii o termínu *digitální stopa*, analyzovat možná nebezpečí a navrhnout vlastní řešení, jak *digitální stopy* monitorovat a jak tomuto monitorování zabránit. Při objasňování tohoto pojmu jsem se dozvěděl, že už jsem u *Národní bezpečnostní agentury (NSA)* veden jako extremista, protože jsem vyhledával slovo *TOR*. Na internetu není téměř nikdo v bezpečí a o každém, kdo internet používá, už je někde uložena jeho *digitální stopa*.

V rámci práce byly vypracovány dvě aplikace. Tou první je vyhledávací robot *robot7*, který lze použít na monitorování *digitálních stop*, přičemž výsledky vyhledávání je možno ukládat a kdykoliv v budoucnu znovu pomocí této aplikace otevřít. Výsledek vyhledávání je nabízen ve dvou formách. Prvním je strom všech odkazů a druhým přehledný graf, na kterém jsou dvě možnosti zobrazení. Při programování tohoto robota jsem zjistil, že vyhledávače nejsou zas tak složité, jak jsem si myslel. Hlavní myšlenka spočívá ve velice sofistikovaném zpracovávání stránky a uložení těchto informací. Samotné vyhledávače už provádí jen rutinní dotazy.

Druhou aplikací je webová aplikace pojmenována *servlet-filter*, která se snaží odhalit dotazujícího se robota a znemožnit mu přístup ke stránkám. Po celou dobu testování mého robota jsem nenarazil na znemožnění stahování stránek ze serveru. To se mi povedlo až za pomocí této mnou naprogramované aplikace, která při možné detekci robota uživateli vrací chybové odpovědi.

V budoucnu lze implementovaného robota určitě rozšířit z mnoha hledisek. Mezi vylepšení by mohla být možnost pracovat s *cookies*, aby mohl prohledávat stránky, na které se uživatel přihlásí. Užitečné by bylo prohledávání souborů (pdf, word, xlsx, atd.) na internetu, které byly nalezeny a v nich vložené odkazy. Robota by bylo možné naprogramovat tak, aby využíval veškerý dostupný výkon. V případě programovací jazyka *Java* by se pro toto dalo použít *fork-join*.

Aplikaci *servlet-filter* by zase bylo možné vylepšit z hlediska detekce robotů. Co když bude vyhledávací robot používat náhodně generovaný čas pro posílaní dotazů na server? Na to už jsou i má řešení nedostačující. V případě mnou implementované aplikace bude vždy záležet na konkrétním aplikačním serveru, jaké nabízí možnosti.

Daniel Žažo

# 8 Reference


[1] Internet live stats: Internet Users. *Internet live stats* [online]. [cit. 2014-10-07]. Dostupné z: http://www.internetlivestats.com/internet-users/

[2] Světem Internetu bez nehod a průšvihů (28.2.2013). *INNET — VŠB - Technická univerzita Ostrava* [online]. [cit. 2014-11-10]. Dostupné z: http://idoc.vsb.cz/cs/okruhy/iit/dokumenty/seminare/20130228/index.html

[3] ČERNÝ, Michal. Digitální stopy a digitální identita. *Digitální stopy a digitální identita* [online]. 2011 [cit. 2014-10-31]. Dostupné z: http://clanky.rvp.cz/clanek/k/g/12943/DIGITALNI-STOPY-A-DIGITALNI-IDENTITA.html/

[4] Digital footprint. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-11-10]. Dostupné z: http://en.wikipedia.org/wiki/Digital_footprint

[5] APNIC history: History of the Regional Internet Registries. ASIA PACIFIC NETWORK INFORMATION CENTRE. [online]. [cit. 2014-11-12]. Dostupné z: http://www.apnic.net/about-APNIC/organization/history-of-apnic/history-of-the-regional-internet-registries

[6] VYSOKÁ ŠKOLA BÁŇSKÁ - TECHNICKÁ UNIVERZITA OSTRAVA. *www.vsb.cz* [online]. © 2014 [cit. 2014-11-13]. Dostupné z: www.vsb.cz

[7] *WHOIS Search, Domain Name, Website, and IP Tools* [online]. © 2014 Who.is [cit. 2014-11-24]. Dostupné z: https://who.is/

[8] VYHLEDÁVÁNÍ V REGISTRU (WHOIS). *CZ.NIC - SPRÁVCE DOMÉNY .CZ* [online]. © 2014 [cit. 2014-11-14]. Dostupné z: http://www.nic.cz/whois/

[9] Jak funguje geolokace ve Firefoxu - detailní (až vyčerpávající) popis. HASSMAN, Martin. *HTML 4 5 6... BLOG ZABÝVAJÍCÍ SE VÝVOJEM HTML A XHTML.* [online]. 2010 [cit. 2014-12-01]. Dostupné z: http://html456.blogspot.cz/2010/06/jak-funguje-geolokace-ve-firefoxu.html

[10] GOOGLE. *Google Maps API Web Services* [online]. 2014 [cit. 2014-12-01]. Dostupné z: https://developers.google.com/maps/documentation/geocoding/

[11] WORLD WIDE WEB CONSORTIUM. *W3C Geolocation API Specification* [online]. 2014 [cit. 2014-12-01]. Dostupné z: http://dev.w3.org/geo/api/spec-source.html

[12] Message Headers. INTERNET CORPORATION FOR ASSIGNED NAMES AND NUMBERS (ICANN). *The Internet Assigned Numbers Authority (IANA) is responsible for the global coordination of the DNS Root, IP addressing, and other Internet protocol resources.* [online]. 2014 [cit. 2014-12-04]. Dostupné z: https://www.iana.org/assignments/message-headers/message-headers.txt

[13] CENTRUM.CZ — ATLAS.CZ 1999 – 2014 © ECONOMIA, a.s. *Centrum.cz* [online]. 1999 – 2014 © [cit. 2014-12-04]. Dostupné z: http://www.centrum.cz/

[14] SEZNAM.CZ, a.s. *Seznam.cz* [online]. Copyright © 1996-2014 [cit. 2014-12-04]. Dostupné z: https://www.seznam.cz/

[15] WORLD WIDE WEB CONSORTIUM. *Http://www.w3.org/* [online]. © 2014 [cit. 2014-12-08]. Dostupné z: http://www.w3.org/

[16] *Anonymous-P2P.org: Anonymous file sharing and communication.* [online]. 2005 [cit. 2014-12-17]. Dostupné z: http://www.anonymous-p2p.org/programs.html

[17] CORE TOR PEOPLE. *Anonymity OnlineProtect your privacy. Defend yourself against network surveillance and traffic analysis.* [online]. 2014 [cit. 2014-12-10]. Dostupné z: https://www.torproject.org/

[18] *COMPUTERWORLD Deník pro IT profesionály: Síť Tor odolává pokusům NSA o průnik* [online]. 05.10.2013. [cit. 2014-12-15]. Dostupné z: http://computerworld.cz/securityworld/sit-tor-odolava-pokusum-nsa-o-prunik-50411

[19] SAHA, Saurabh. Top 100 Free Proxy Sites – Free Proxy Servers List 2014. *TechGYD - Technology Blog* [online]. Jul 27, 2014 [cit. 2014-12-15]. Dostupné z: http://www.techgyd.com/free-proxy-sites-list-2014/12890/

[20] Jaký je trest za trestný čin pomluvy. *BEZPLATNÁ PRÁVNÍ PORADNA* [online]. Copyright © Eva Mlčochová 2009 - 2015 [cit. 2015-01-09]. Dostupné z: http://bezplatnapravniporadna.cz/online-zdarma/trestni-pravo/tresty-a-ochranna-opatreni/2253-jaky-je-trest-za-trestny-cin-za-pomluvy.html

[21] *Facebook: Connect with friends and the world around you on Facebook.* [online]. Facebook © 2015 [cit. 2015-01-07]. Dostupné z: https://www.facebook.com/

[22] *Google+: One account. All of Google.* [online]. ©2015 Google [cit. 2015-01-07]. Dostupné z: https://accounts.google.com/

[23] *Welcome to Twitter.: Connect with your friends — and other fascinating people. Get in-the-moment updates on the things that interest you. And watch events unfold, in real time, from every angle.* [online]. © 2015 Twitter, Inc. [cit. 2015-01-07]. Dostupné z: https://twitter.com/?lang=en

[24] *Prohlášení o právech a povinnostech* [online]. 15. listopadu 2013 [cit. 2015-01-07]. Dostupné z: https://cs-cz.facebook.com/legal/terms

[25] *PLEASE ROB ME: Raising awareness about over-sharing* [online]. Copyright © 2010 [cit. 2015-01-09]. Dostupné z: http://pleaserobme.com/

[26] *FORSQUARE: Introducing the all-new Foursquare, which learns what you like and leads you to places you'll love.* [online]. Foursquare © 2015 [cit. 2015-01-09]. Dostupné z: https://foursquare.com/

[27] FORD, Melissa. *BlogHer: Please Rob Me and How the Internet Affects Privacy* [online]. March 04, 2010 [cit. 2015-01-10]. Dostupné z: http://www.blogher.com/please-rob-me-and-how-internet-affects-privacy

[28] RYAN, Thomas. "Getting In Bed with Robin Sage.". In: *"Getting In Bed with Robin Sage."* [online]. ©2010 Provide Security, LLC [cit. 2015-01-12]. Dostupné z: http://media.blackhat.com/bh-us-10/whitepapers/Ryan/BlackHat-USA-2010-Ryan-Getting-In-Bed-With-Robin-Sage-v1.0.pdf

[29] Robin Sage. *From Wikipedia, the free encyclopedia* [online]. 24 July 2014 [cit. 2015-01-12]. Dostupné z: http://en.wikipedia.org/wiki/Robin_Sage

[30] EBAY INC. *EBay* [online]. Copyright © 1995-2015 [cit. 2015-01-12]. Dostupné z: http://www.ebay.com/

[31] Odmítl prodat Googlu svůj nápad za tři čtvrtě miliardy. Teď na něm sám vydělává. *First Class: Svět a myšlení úspěšných* [online]. 20.3.2014 [cit. 2015-01-12]. Dostupné z: http://www.firstclass.cz/2014/03/odmitl-prodat-googlu-svuj-napad-za-tri-ctvrte-miliardy-ted-na-nem-sam-vydelava/#sthash.JkwV1KGY.nzfkNxcQ.dpbs

[32] CZ.NIC, z. s. p. o. *MojeID: Internet bez hesel a registrací* [online]. Copyright 2015 [cit. 2015-01-19]. Dostupné z: https://www.mojeid.cz

[33] CZ.NIC, z. s. p. o. *MojeID: Internet bez hesel a registrací* [online]. Copyright 2015 [cit. 2015-01-19]. Dostupné z: https://www.mojeid.cz

[34] Robots Database. *The Web Robots Pages* [online]. © 2007 [cit. 2015-04-07]. Dostupné z: http://www.robotstxt.org/db.html

[35] Googlebot. *Webmaster Tools Help* [online]. ©2015 Google [cit. 2015-04-07]. Dostupné z: https://support.google.com/webmasters/answer/182072

[36] Top 10 Bots You Should Know About. GAFFAN, Marc. *Incapsula's Blog* [online]. 2012-08-21 [cit. 2015-04-07]. Dostupné z: https://www.incapsula.com/blog/know-your-top-10-bots.html

[37] Internetový bot. *From Wikipedia, the free encyclopedia* [online]. 2015-02-05 [cit. 2015-04-08]. Dostupné z: http://cs.wikipedia.org/wiki/Internetov%C3%BD_bot

[38] What is Java?. ORACLE. *What is Java?* [online]. 2015 [cit. 2015-04-08]. Dostupné z: http://java.com/en/download/whatis_java.jsp

[39] Eclipse Is... An amazing open source community of Tools, Projects and Collaborative Working Groups. Discover what we have to offer and join us. *Eclipse* [online]. © 2015 [cit. 2015-04-08]. Dostupné z: `https://eclipse.org/`

[40] Welcome to Apache Maven. THE APACHE SOFTWARE FOUNDATION. © *2002-2015* [online]. © 2002-2015 [cit. 2015-04-08]. Dostupné z: `http://maven.apache.org/`

[41] Seznamte se – DoS a DDoS útoky. *Security-Portal.cz: we separate geeks from kiddies* [online]. 2013 [cit. 2015-04-16]. Dostupné z: `http://www.security-portal.cz/clanky/seznamte-se-%E2%80%93-dos-ddos-%C3%BAtoky`

[42] About /robots.txt. *GET /ROBOTS.TXT* [online]. 2007 [cit. 2015-04-16]. Dostupné z: `http://www.robotstxt.org/robotstxt.html`

[43] Apache Tomcat. *Apache Tomcat* [online]. © 1999-2015 [cit. 2015-04-18]. Dostupné z: `http://tomcat.apache.org/`

Přílohy

# A Obsah DVD

Obsah jednotlivých adresářů obsažených v přiloženém DVD.

- **bin** - obsahuje spustitelné verze vyvíjeného softwaru
  - **robot7.jar** - spustitelný *JAR* soubor vyhledávacího robota
  - **servlet-portlet.war** - balíček určen pro serverovou část na ochranu *digitálních stop* před monitorováním
- **src** - obsahuje zdrojové kódy vyvíjeného softwaru
  - **robot7**
  - **servlet-portlet**
- **support** - obsahuje doprovodný software
  - **apache-tomcat-7.0.61.zip** - aplikační server, na němž funguje *servlet-portlet*
- **text** - obsahuje text diplomové práce
  - **robot7&servlet-filter.pdf** - text diplomové práce
  - **Latex - SourceCode** - Zdrojové soubory *Latex* pro vygenerování výsledného *PDF souboru*

# B Uživatelský manuál

1. Po spuštění *robota7* je výchozí *URL adresa* nastavena na *localhost* a to přímo na aplikaci *servlet-filter*. Je-li tedy nainstalována, tak se může hned otestovat. V opačném případě uživatel zadá svou *URL adresu*, na které chce začít vyhledávat.

2. Na úvodní záložce *Search criteria* může uživatel specifikovat následující údaje pro vyhledávání.

   - **Default URL** - Výchozí bod pro vyhledávání.
   - **Maximum depth** - Maximální hloubka, ve které se má vyhledávat.
   - **Maximum pages** - Maximální počet stránek, které mají být prohledány.
   - **Searched domains** - Domény, na kterých se má vyhledávat.
   - **Searched words** - Vyhledávaná slova. Pokud chce uživatel vyhledávat více slov, tak jedno slovo musí být na jednom řádku.
   - **Search only default host** - V případě zaškrtnutí této možnosti jsou prohledávané stránky na stejném *hostu* jako je výchozí *URL adresa*.
   - **Set timeout for search** - Touto možnosti uživatel může nastavit *timeout* pro jednotlivé dotazy na prohledávané stránky.

3. Na všech záložkách je viditelný průběh vyhledávání v podobě následující statistiky.

   - **Found links** - Počet všech nalezených odkazů (i špatných). To, že se našel odkaz ještě neznamená, že daná stránka pod tímto odkazem existuje. To je zohledněno statistikou *Wrong links*.
   - **Wrong links** - Počet odkazů, které nebylo možné prohledat. Důvod neúspěchu lze zjistit na záložce *Error log*.
   - **Pages with found phrases** - Hledá-li uživatel konkrétní slova na stránkách, tak zde je počet stránek, na kolika stránkách bylo aspoň jedno hledané slovo nalezeno. Vyhledávání není *case sensitive*.
   - **Time** - Celkový čas vyhledávání.

4. Na všech záložkách jsou viditelné taky tlačítka spolu se *status barem*.

   - **Start searching** - Spuštění vyhledávání.
   - **Stop searching** - Ukončení vyhledávání.
   - **Exit application** - Ukončení celé aplikace. Pokud běží vyhledávání, tak je zrušeno a aplikace se ihned ukončí.
   - **Status bar** - Podle *status baru* by se měl vždy každý uživatel řídit. Dokud je v pohybu, tak aplikace pracuje. V případě stopnutí (tlačítkem *Stop searching*) se aplikace nemusí ukončit ihned. Aby bylo vyhledávání korektně zastaveno, tak se musí uvnitř aplikace provést několik důležitých kroků a to nemusí být ihned.

5. Po spuštění vyhledávání se aplikace automaticky přepne na záložku *Result tree*. Zde jsou postupně přidávány odkazy do stromu. Vyhledává-li uživatel nějaké slova a to slovo bylo na nějakém odkazu (stránce) nalezeno, tak se navíc pozadí odkazu v tomto stromu obarví nazeleno. Dvojklikem lze daný odkaz otevřít v uživatelově výchozím prohlížeči.

6. Na záložce *Error log* uživatel vidí, proč nejsou nalezené linky přidány do výsledku.

7. Na poslední záložce *Graph* uživatel může tlačítkem *Create graph in new window* vytvořit graf z nalezených odkazů. Tento graf se otevře v novém okně, kde má uživatel další možnosti týkající se grafu. Pokud už uživatel v minulosti vytvořil graf a uložil jej, tak druhým tlačítkem *Open graph* na této stránce může uložený graf zobrazit.

8. Nechá-li si uživatel vytvořit graf, otevře se nové okno s grafem, které obsahuje následující možnosti.

   - **Switch layout** - Přepínaní rozvržení grafu. Aktivní rozvržení má zelené pozadí. (*TREE* nebo *BALLOON).*
   - **Zooming** - Přibližování a oddalování grafu.
   - **Mouse mode** - Slouží k výběru chování grafu. Pokud je aktivní výchozí hodnota *TRANSFORMING*, tak uživatel stisknutím levého tlačítka myši a následným pohybem pohybuje celým grafem. Je-li vybrána druhá varianta *PICKING*, tak uživatel může přesouvat uzly grafu dle libosti.
   - **Hyperbolic view** - Zobrazení hyperbolického náhledu, které slouží jako lupa.
   - **Export** - Vyexportování grafu do souboru *xml*. Uživatel vždy zadává jméno souboru ručně.
   - **Close** - Zavření celého okna s grafem.

9. Instalace webové aplikace *servlet-filter* se provede poměrně jednoduše. Prvně je potřeba rozbalit soubor *apache-tomcat-7.0.61.zip*, kde se nachází aplikační server *Apache Tomcat*.V případě operačního systému *Windows* se *Tomcat* spustí souborem *startup.bat*. V případě *linuxu* se spustí souborem *startup.sh*. Oba soubory se nachází v domovském adresáři *Tomcatu* ve složce *bin*.
   Nakonec je potřeba aplikaci *servlet-filter.war* nakopírovat do složky *webapps*, která se rovněž nachází v domovském adresáři *Tomcatu*. V logu lze sledovat celý průběh. Aplikace bude dostupná na *URL adrese* `http://127.0.0.1:8080/servlet-filter/`

10. Ukončení *Tomcatu* se provede souborem *shutdown.bat* nebo *shutdown.sh* v závislosti na prostředí.